BRAIN architecture specifications and models, BRAIN functionality and protocol specification

Alberto Lopez Toledo , Hector Velayos, Nuria Villaseor, Phil Eardley, Louise Burness, Dave Wisely, David Higgins, Severine Corairie, Lorenzo Venturato, Youssef Fadili, Gosta Leijonhufvud, Ands Gbor Valk, Csaba Keszei, Zoltn Turnyi, Philippe Bertin, Karine Guillouard, Servane Bonjour, Youssef Khouaja, Nikos Georganopoulos, Andrej Mihailovic, Ian Groves, Mika Liljeberg, Tapio Suihko, Markku Kojo, Jukka Manner, Aki Laukkanen, Robert Hancock, Eleanor Hepworth, Mark West, Andreas Kassler, Ern Kovacs, Oliver Schram, Rdiger Geib.*

Abstract: This document presents the final research output of the BRAIN projects Work Package 2, which covers the BRAIN Access Network. The topic areas addressed are: Network Layer Architecture, IP Mobility Management, Quality of Service, Enhanced Socket Interface and IP2W Interface. Detailed analysis of various proposals is included, together with candidate protocols and extensions that we have developed. Results are given from simulations that support and validate some of the described aspects, and a worked example is presented that illustrates how the architecture and protocol work could fit together to solve a users needs. Open issues are identified and their impact is discussed. The Deliverable also serves as input for the followon project, MIND, which will implement and test our proposed solutions to the key issues.

Keywords: BRAIN, WP2, A2.3, A2.4, BRAIN Access Network Architecture, Conclusions, Protocol models and specifications, Mobility management, Quality of Service, Interfaces, Example

*Please see inside for affiliation of all the authors

IST-1999-10050 BRAIN

D2.2

BRAIN architecture specifications and models, BRAIN functionality and protocol specification

Contractual Date of Delivery to the CEC:	**31/03/01**
Actual Date of Delivery to the CEC:	**30/03/01**
Author(s):	WP2
Participant(s):	SM, BT, ASSA, ERA, FTR&D, INRIA, KCL, NOK, NTT DoCoMo, SONY, T-NOVA
Workpackage:	WP2 – Access Network Architecture
Est. person months:	80
Security:	Public
Nature:	**R**
Version:	1.0
Total number of pages:	425

Abstract:

This document presents the final research output of the BRAIN project's Work Package 2, which covers the BRAIN Access Network. The topic areas addressed are: Network Layer Architecture, IP Mobility Management, Quality of Service, Enhanced Socket Interface and IP2W Interface. Detailed analysis of various proposals is included, together with candidate protocols and extensions that we have developed. Results are given from simulations that support and validate some of the described aspects, and a 'worked example' is presented that illustrates how the architecture and protocol work could fit together to solve a user's needs. Open issues are identified and their impact is discussed. The Deliverable also serves as input for the follow–on project, MIND, which will implement and test our proposed solutions to the key issues.

Keyword list: **BRAIN, WP2, A2.3, A2.4, BRAIN Access Network Architecture, Conclusions, Protocol models and specifications, Mobility management, Quality of Service, Interfaces, Example**

Dedication

The authors would like to dedicate this deliverable to Aki Laukkanen, a much liked and highly respected colleague who made many valuable contributions to our work before his tragic death earlier this year.

Our sympathies go out to his family and friends in Finland in their grief

Executive Summary

The BRAIN (Broadband Radio Access for IP Based Networks) IST project is a wide-ranging research activity that aims to design and specify an IP-based mobile wireless access system that will complement 2nd and 3rd generation mobile systems. It encompasses the whole system from top to bottom. This report, Deliverable D2.2, focuses on the network layer which supports and unifies the whole system. Other reports from the project cover mobility-aware user applications and middleware (D1.2) and broadband radio access through HIPERLAN/2 (D3.2). Overall, the target of the BRAIN project is a complete system that provides seamless support of IP-based services for users in hot spots and on the move.

This Deliverable discusses the design of the BRAIN access network architecture. It presents a critical analysis of mobility management and quality of service within the BRAIN access network, and includes a description of candidate protocols and extensions that we have developed. It describes two network-layer interfaces that we have specified: one towards wireless link layers and the other towards applications / middleware.

D2.2 is structured in two parts: the first comprises a short, "core" report; and the second, larger part consists of a series of more extensive Annexes.

The 'Core' presents our key conclusions on the main topic areas, and briefly justifies them. The Annexes complement and support the Core, and contain detailed analyses and specifications. There are numerous references within the Core to the Annexes, and it is suggested that the reader begins with the Core, before exploring particular topics in more detail via the appropriate Annex.

The 'Core' consists of the following Sections:

?? The first section is an introduction to the deliverable D2.2, including a realistic user scenario that motivates the need for work on the BRAIN access network.

?? Next, the BRAIN access network architecture is described, demonstrating how the various parts interwork. Included here is a discussion of our approach to security, global mobility, and BRAIN – UMTS interoperation.

?? The research conclusions on IP mobility management are presented next, addressing issues such as handover management, path updates, paging, scalability and resilience; these are followed by a novel candidate solution for micro-mobility.

?? In the section on Quality of Service, a standards-based baseline QoS architecture is described, followed by extensions that can be used to optimise the QoS solution for particular scenarios.

?? BRAIN interfaces are next presented. The Enhanced Socket Interface (ESI) and the IP2W (IP to Wireless) interface are designed to connect the BRAIN network layer with the application and link layer in a manner to provide complete QoS support.

?? The final section of the "core" report presents one example of how the preceding architecture and protocol work could fit together to solve a user's needs.

Authors

Partner	Name	Phone / Fax / E-mail
Agora Systems/UPM	Alberto Lopez	Phone: +34 91 5495700 x 442 Fax: +34 91 3367333 E-mail: alberto@dit.upm.es
Agora Systems	Hector Velayos	Phone: +34 91 533 58 57 Fax: +34 91 534 84 77 E-mail: hector_velayos@agora-systems.com
Agora Systems	Nuria Villaseñor	Phone: +34 91 533 58 57 Fax: +34 91 534 84 77 E-mail: nuria_villasenor@agora-systems.com
BTexaCT	Phil Eardley	Phone: +44 1473 645938 Fax: +44 1473 646885 E-mail: philip.eardley@bt.com
BTexaCT	Louise Burness	Phone: +44 1473 646504 Fax: +44 1473 646885 E-mail: louise.burness@bt.com
BTexaCT	Dave Wisely	Phone: +44 1473 643848 Fax: +44 1473 646885 E-mail: dave.wisely@bt.com
BTexaCT	David Higgins	Phone: +44 1473 645232 Fax: +44 1473 646885 E-mail: david.j.higgins@bt.com
BTexaCT	Severine Corairie	Phone: contact Phil Eardley (details above) Fax: E-mail:
BTexaCT	Lorenzo Venturato	Phone: +44 1473 645028 Fax: +44 1473 646885 E-mail: lorenzo.venturato@bt.com
BTexaCT	Youssef Fadili	Phone: contact Phil Eardley (details above) Fax: E-mail:

Ericsson	Gosta Leijonhufvud	Phone: Fax: E-mail:	+46 8 4047243 +46 8 50878090 gosta.leijonhufvud@era.ericsson.se
Ericsson	Andás Gábor Valkó	Phone: Fax: E-mail:	+36 1 437 7774 +36 1 437 7219 andras.valko@eth.ericsson.se
Ericsson	Csaba Keszei	Phone: Fax: E-mail:	+36 1 437 4901 +36 1 437 7219 csaba.keszei@eth.ericsson.se
Ericsson	Zoltán Turányi	Phone: Fax: E-mail:	+36 1 437 7636 +36 1 437 7219 andras.valko@eth.ericsson.se
France Télécom R&D	Philippe Bertin	Phone: Fax: E-mail:	+33 2 99124157 +33 2 99124098 philippe.bertin@rd.francetelecom.fr
France Télécom R&D	Karine Guillouard	Phone: Fax: E-mail:	+33 2 99124724 +33 2 99124098 karine.guillouard@rd.francetelecom.fr
France Télécom R&D	Servane Bonjour	Phone: Fax: E-mail:	+33 2 99 12 46 84 +33 2 99 12 40 98 servane.bonjour@rd.francetelecom.fr
France Télécom R&D	Youssef Khouaja	Phone: Fax: E-mail:	+33 2 99123840 +33 2 99124098 youssef.khouaja@rd.francetelecom.fr
France Télécom R&D	Olivier Charles	Phone: Fax: E-mail:	+33 1 45296770 +33 1 45296519 olivier.charles@rd.francetelecom.fr
Kings College London	Hamid Aghvami	Phone: Fax: E-mail:	+44 20 78482898 +44 20 78482664 hamid.aghvami@kcl.ac.uk
Kings College London	Nikos Georganopoulos	Phone: Fax: E-mail:	+44 20 78482889 +44 20 78482664 nikolaos.georganopoulos@kcl.ac.uk

Kings College London	Andrej Mihailovic	Phone: Fax: E-mail:	+44 20 78482889 +44 20 78482664 andrej.mihailovic@kcl.ac.uk
Kings College London	Ian Groves	Phone: Fax: E-mail:	+44 1473 257344 +44 1473 218334 ian.groves@kcl.ac.uk
Nokia	Mika Liljeberg	Phone: Fax: E-mail:	+358 40 7431989 +358 9 43766850 mika.liljeberg@nokia.com
Nokia (VTT Information Technology)	Tapio Suihko	Phone: Fax: E-mail:	+358 9 4566078 +358 9 43766850 ext-tapio.suihko@nokia.com
Nokia (University of Helsinki)	Markku Kojo	Phone: Fax: E-mail:	+358 9 19144179 +358 9 19144441 markku.kojo@cs.helsinki.fi
Nokia (University of Helsinki)	Jukka Manner	Phone: Fax: E-mail:	+358 9 19144210 +358 9 19144441 jukka.manner@cs.helsinki.fi
Nokia (University of Helsinki)	Aki Laukkanen	Phone: Fax: E-mail:	
NTT-DoCoMo	Yasushi Yamao	Phone: Fax: E-mail:	+81 468 403166 +81 468 403798 yamao@mlab.yrp.nttdocomo.co.jp
Siemens (Roke Manor Research)	Robert Hancock	Phone: Fax: E-mail:	+44 1794 833601 +44 1794 833434 robert.hancock@roke.co.uk
Siemens (Roke Manor Research)	Eleanor Hepworth	Phone: Fax: E-mail:	+44 1794 833146 +44 1794 833434 eleanor.hepworth@roke.co.uk
Siemens (Roke Manor Research)	Mark West	Phone: Fax: E-mail:	+44 1794 833311 +44 1794 833434 mark.a.west@roke.co.uk

Siemens (University of Ulm)	Andreas Kassler	Phone: Fax: E-mail:	+49 731 5024139 +49 731 5024142 kassler@informatik.uni-ulm.de

Sony	Ernö Kovacs	Phone: Fax: E-mail:	+49 711 5858298 +49 711 5858468 kovacs@sony.de

Sony	Oliver Schram	Phone: Fax: E-mail:	+49 711 5858798 +49 711 5858468 schramm@sony.de

T-Nova	Rüdiger Geib	Phone: Fax: E-mail:	+49 6151 83 2138 +49 6151 83 8103 ruediger.geib@telekom.de

Table of Contents

1 Introduction 23

1.1 Introduction to the Deliverable 23
1.2 Structure and Scope of the Deliverable 23
1.3 Prelude – BRAIN Network Motivation, a User Perspective 24

2 The BRAIN Network Architecture 26

2.1 Why An IP Access Network 26
2.2 Design Principles – "What IP-Based Really Means" 27
2.3 The Access Network Problem Definition 28
2.3.1 Addressing and Routing 28
2.3.2 Scalability and Resilience 29
2.3.3 Security 30
2.3.4 Radio Resource Management 30
2.4 External Interactions 31
2.4.1 Global IP Mobility 31
2.4.2 UMTS and Other Mobile Networks 31
2.5 Access Network Components 32

3 IP Mobility Management 33

3.1 Introduction 33
3.1.1 Overall Approach and Interactions of Key Functions 33
3.2 Handover Management 34
3.2.1 Handover Problem 34
3.2.2 Existing Solutions 34
3.2.3 Conclusions on Handover Management 35
3.3 Path Updates 36
3.3.1 The Path Update Problem 36
3.3.2 Existing Solutions 37
3.3.3 Attributes of Solutions 38
3.3.4 Conclusion 38
3.4 Scalability and Resilience 38
3.4.1 The Case for Multiple BMGs 39
3.4.2 Routing 39
3.5 Paging 40
3.5.1 Introduction – Idle and Stand-by Modes 40
3.5.2 Existing Solutions to Paging 40
3.5.3 BRAIN Paging proposal 40
3.6 BRAIN Candidate Mobility Protocol 42
3.6.1 Introduction 42
3.6.2 Key features of BRAIN Candidate Mobility Protocol 42
3.6.3 Discussion of BRAIN Candidate Mobility Protocol 43

4 Quality of Service 45

4.1 Introduction 45
4.2 Base-Line Architecture 46
4.2.1 Basic Design Choices 46
4.2.2 Description of Base-Line Architecture 47
4.2.3 Error Reporting 48
4.2.4 Evaluation 48
4.3 Solutions to weaknesses in the base-line architecture 50
4.3.1 QoS Context Transfer 50
4.3.2 Bandwidth Broker 50
4.3.3 Hop-by-hop Call Admission 51
4.3.4 Bounded Delay DS 52
4.3.5 Simplified service definitions and signalling protocols 52
4.3.6 Mobility Enhanced QoS Parameters 53

4.3.7 Hard State RSVP 53
4.3.8 QoS Reservations in Temporary Tunnels 53
4.3.9 DS Handover markings 53
4.3.10 Local BAN signalling Protocol 54
4.3.11 RSVP Proxies 54
4.4 Extending the Base-line architecture 55
4.4.1 The strengths of service guarantees that can be achieved are limited because of the call admission architecture 56
4.4.2 Minimize the quantity of signalling required 56
4.4.3 Seamless handover 56
4.4.4 Reservation QoS in the absence of end-to-end QoS signalling 57
4.4.5 Discussions 57

5 BRAIN Interfaces 58

5.1 Introduction 58
5.2 Enhanced Socket Interface 59
5.2.1 Design Principles 59
5.2.2 Design Decisions 59
5.2.3 ESI Primitives 59
5.2.4 QoS and Primitive Mapper 60
5.2.5 Legacy Application Support 60
5.2.6 Local Management Functionality 61
5.2.7 Multi-homing 61
5.3 IP to Wireless Interface 61
5.3.1 Handover Support 62
5.3.2 Idle Mode Support 64
5.3.3 QoS Support 65

6 Worked example 68

6.1 Introduction 68
6.2 Registration 68
6.3 Service example 1 - Multicast 70
6.4 Service example 2 - Voice 71
6.5 Handover 72
6.6 Paging 73
6.7 Conclusions 74

7 References 75

A1 BRAIN WP2 ANNEX 76

A2 Architecture Annex 77

A2.1 Modular IP Architectures For Wireless Mobile Access 77
A2.1.1 Abstract 77
A2.1.2 Introduction 77
A2.1.3 Motivation for the BRAIN Access Network 78
A2.1.4 Overall Structure and Requirements for the Access Network 83
A2.1.5 An Outline Design for the Access Network 89
A2.1.6 System Design for Key Features 92
A2.1.7 Acknowledgements 96
A2.1.8 Paper References 97
A2.2 Mobility Related Terminology 98
A2.2.1 IETF Draft 98
A2.3 BRAIN – UMTS Interoperation 116
A2.3.1 Coupling 116
A2.3.2 Other Issues/ Discussions 122
A2.3.3 BRAIN - UMTS Interoperation References 124
A2.4 Security 125
A2.4.1 Introduction 125

A2.4.2 Security Objectives 125
A2.4.3 AAA 126
A2.4.4 Trust relationship 132
A2.4.5 BAR Security Functions 134
A2.4.6 Encryption on the radio link 134
A2.4.7 Authentication between the MN, the BAN and the home network 136
A2.4.8 How to get the material to sign/check: keys, certificates, etc…? 137
A2.4.9 Security References 137
A2.5 Diversity Combining and Soft Handover Support 138
A2.5.1 Background 138
A2.5.2 Macro and Micro diversity. 139
A2.5.3 Macro-diversity in BRAIN 140
A2.5.4 Why IP diversity combining is difficult 140
A2.5.5 Diversity References 142
A2.6 Radio Resource Management 143
A2.6.1 Protocol Architecture 143
A2.7 Multicast 145
A2.7.1 Introduction to Multicast 145
A2.7.2 Mobility and Multicast 146
A2.7.3 Multicast References 146
A2.8 Location Based Service Support 147
A2.8.1 Application scenarios 147
A2.8.2 Requirements 148
A2.8.3 Non-Requirements 149
A2.8.4 Existing approaches 149
A2.8.5 Location Based Services References 150

A3 Mobility Management Annex 151

A3.1 IETF Handover Protocols and BRAIN Handover Design 151
A3.1.1 Introduction 151
A3.1.2 Generalised Handover Framework 151
A3.1.3 Proposed Handover Schemes 153
A3.1.4 Comparison of the Handover Schemes 160
A3.1.5 BRAIN Handover Protocol Design 164
A3.1.6 Conclusions about a Generic Handover Protocol 167
A3.1.7 Handover Management References 169
A3.2 Paging support 170
A3.2.1 Basic conclusions for a paging mechanism supported in the BAN 170
A3.2.2 Existing proposals 170
A3.2.3 Recommendations for a BRAIN paging scheme 176
A3.2.4 Paging scheme with the BRAIN Candidate Mobility Protocol 179
A3.2.5 Paging References 180
A3.3 Path Updates 182
A3.3.1 General 182
A3.3.2 Topologies 182
A3.3.3 Issues 183
A3.3.4 Interactions 185
A3.3.5 Solutions 187
A3.3.6 Path Updates References 188
A3.4 Scalability and Resilience 189
A3.4.1 Failure of Nodes 189
A3.4.2 Scalability 190
A3.4.3 Multiple BRAIN Mobility Gateways 191
A3.4.4 Data flows 193
A3.4.5 Inter-BMG handover 194
A3.4.6 Multi-BMG Routing 195
A3.4.7 BMG Routing 197
A3.4.8 BRAIN Candidate Mobility Protocol 197
A3.4.9 Scalability and Resilience References 198
A3.5 BRAIN Candidate Mobility Protocol Components 199

A3.5.1 Initial Login 199
A3.5.2 Handover Preparation 200
A3.5.3 Handover Execution 201
A3.5.4 Change Anchor Point 202
A3.5.5 Logout 203
A3.5.6 Paging Mechanism 204
A3.6 Mobility Management and QoS in BRAIN Access Networks (London Workshop Paper) 206
A3.6.1 Abstract 206
A3.6.2 Introduction 206
A3.6.3 Mobility Management 207
A3.6.4 Micro mobility and QoS 214
A3.6.5 Conclusion 219
A3.6.6 Paper References 219
A3.7 A Framework for the Evaluation of IP Mobility Protocols (PIMRC paper) 222
A3.7.1 Abstract 222
A3.7.2 Introduction 222
A3.7.3 Classification of IP Mobility protocols 223
A3.7.4 Evaluation Framework 224
A3.7.5 Initial comparison of IP-mobility proposals using our Evaluation Framework 226
A3.7.6 Conclusions 230
A3.7.7 Acknowledgement 231
A3.7.8 Paper References 231

A4 Quality of Service Annex 233

A4.1 Introduction 233
A4.2 Summary of QoS Interactions 233
A4.2.1 QoS and Higher Level Protocols 233
A4.2.2 Possible Interaction between L2 and L3 QoS mechanisms 234
A4.2.3 Wireless Issues 235
A4.2.4 Mobility Issues 236
A4.3 Base Line Architecture 250
A4.3.1 Basic Design Choices 250
A4.3.2 Description of Base Line Architecture in depth 252
A4.3.3 Error Reporting 257
A4.3.4 Evaluation 257
A4.4 Solutions to Weaknesses in the Base-Line Architecture 262
A4.4.1 QoS Context Transfer 262
A4.4.2 Bandwidth Broker 263
A4.4.3 Coupling of hop-by-hop call admission with Current Micro-Mobility Mechanisms 264
A4.4.4 Repairing RSVP local Path repair 267
A4.4.5 Bounded Delay Service 269
A4.4.6 Simpler QoS Classes 270
A4.4.7 Mobility Enhanced QoS Parameters - Generic IP QoS signalling interfaces in a mobile environment 271
A4.4.8 Optimised RSVP 273
A4.4.9 Local BAN signalling Protocol 274
A4.4.10 RSVP Proxies 276
A4.5 Discussion Topics 279
A4.5.1 Protection of the mobile terminal 279
A4.5.2 Candidate Handover Node Selection 279
A4.5.3 QoS and the BCMP 279
A4.5.4 QoS Interaction with Tunnels 282
A4.5.5 Combined L2/3 signalling 283
A4.6 Quality of Service References 284

A5 Enhanced Socket Interface Annex 287

A5.1 Introduction 287
A5.2 Quality of Service 288
A5.2.1 QoS Contract 288

A5.2.2 The concept of a QoS Service Provider 289
A5.3 ESI from a BRENTA point of view 290
A5.4 Modelling an Interface 291
A5.5 Design Principles 293
A5.6 Enhanced Service Layer 294
A5.7 Design Decisions 295
A5.8 Enhanced Socket Interface 296
A5.8.1 SetQoS, a confirmed service of the ESL 296
A5.8.2 SetQoSViolation notification 298
A5.8.3 AssignQoS, an unconfirmed of the ESL 299
A5.8.4 QoSViolation Notification 300
A5.8.5 ChangeQoS, a confirmed service of the ES 301
A5.8.6 ChangeQoSViolation Notification 303
A5.8.7 ReleaseQoS 304
A5.8.8 Summary 305
A5.8.9 Usage of the ESI primitives 306
A5.9 Brain SI 312
A5.9.1 BRAIN Connection-oriented Service Provider 312
A5.9.2 BRAIN Connection-less Service Provider 312
A5.9.3 BRAIN QoS Service Provider 313
A5.10 QoS Parameters for the ESI 317
A5.10.1 QoS Parameters 317
A5.10.2 FlowSpec 318
A5.10.3 Service Provider specific Information [optional] 319
A5.11 QoS and Primitive Mapper 322
A5.11.1 Primitive Mapper 322
A5.11.2 QoS Mapper 324
A5.12 Support of QoS unaware Application 327
A5.12.1 What is a legacy Application? 327
A5.12.2 Consider the support of Receiver QoS unaware Applications 328
A5.12.3 How can a Configurator be supported 328
A5.12.4 Additional ESI primitives for supporting QoS unaware Applications 328
A5.13 Local Management Functionality 334
A5.13.1 Background 334
A5.13.2 Management Model 334
A5.13.3 BRAIN Management Functions 335
A5.14 Analyse Mobility Management-related aspects of the ESI 337
A5.15 Appendix 339
A5.15.1 Terminology 339
A5.15.2 Explicit Congestion Notification 339
A5.15.3 Requests from Work Group 1 339
A5.16 Enhanced Service Interface References 344

A6 IP2W Interface Annex 345

A6.1 IP_2W Convergence Model 345
A6.1.1 Overview 345
A6.1.2 Address Management 347
A6.1.3 Quality-of-Service Control 351
A6.1.4 Handover Control 358
A6.1.5 Idle Mode Support 362
A6.1.6 Security 365
A6.1.7 Configuration 366
A6.1.8 Data Transfer 368
A6.2 Interface Specification 369
A6.2.1 About Primitive Notation 369
A6.2.2 Data Structures and code values 369
A6.2.3 Address Management Interface 370
A6.2.4 Quality-of-Service Control Interface 373
A6.2.5 Association and Handover Control Interface 378
A6.2.6 Idle Mode Interface 383

A6.2.7 Security Management Interface 384
A6.2.8 Configuration Interface 386
A6.2.9 Data Interface 389
A6.3 References 393

A7 Simulations Annex 394

A7.1 Introduction 394
A7.1.1 Context 394
A7.1.2 Scope 394
A7.1.3 Relationship to Other Activities and Workpackages 395
A7.1.4 Structure 395
A7.2 Simulation tools 396
A7.2.1 Introduction 396
A7.2.2 ns-2 396
A7.3 Simulation Framework 399
A7.3.1 Network Topology 399
A7.4 Simulation Studies and Results 405
A7.4.1 BRAIN Candidate Mobility Protocol 405
A7.4.2 Hawaii simulations 412
A7.4.3 Hawaii and RSVP Coupling Simulations 417
A7.5 References 425

List of Figures

Figure 2-1: Scope of the BRAIN Network Layer....26
Figure 2-2: Network Transparency....27
Figure 2-3: BRAIN Addressing Model....29
Figure 2-4: Network Scalability....29
Figure 2-5: Security Associations....30
Figure 3-1: Relationship of Main Functions of IP Mobility Solution....34
Figure 3-2: BRAIN Planned Handover....36
Figure 3-3: BRAIN Unplanned Handover....36
Figure 3-4: Mobility protocol convergence....37
Figure 3-5: A BAN structure with multiple BMGs....39
Figure 3-6: Paging is initiated from the last BAR that the MN was active on....41
Figure 3-7: Outline of the BRAIN Candidate Mobility Protocol in a BAN....42
Figure 3-8: Snapshot of Simulation of BCMP- Planned HO....44
Figure 4-1: QoS in the Network Nodes....47
Figure 4-2: Simulation Results- Delay of VoIP Packets During a HO when De-coupled (left) and Loosely Coupled (right)....51
Figure 4-3: RSVP Path Repair....52
Figure 4-4: Handover Markings....54
Figure 4-5: Use of RSVP Proxies....55
Figure 5-1: Brain Mobile Terminal Stack....58
Figure 5-2: Availability of ESI, LMI and IP2W....58
Figure 5-3: IP_2W Interface Model....62
Figure 5-4: Planned Handover....64
Figure 5-5: Paging Request at the Mobile Nodes....65
Figure 5-6: Various Scheduling Model Choices....66
Figure 6-1: General Architecture of the University Network....69
Figure 6-2: Message Flows for Multicast Example....70
Figure 6-3: Message Flows for Voice Connection Across the BAN....71
Figure 6-4: Handover Message Flow....72
Figure 6-5: Paging Message Flow (Change of Paging Area)....73
Figure 6-6: Paging Message Flow (Paging Request)....74
Figure A2-1: The Network Layer in the Context of BRAIN....78
Figure A2-2: The Transparent Network....81
Figure A2-3: Application of a Common Air Interface Signalling Protocol....83
Figure A2-4: End-to-End Address Assignments....84
Figure A2-5: Access Network Scalability....85
Figure A2-6: Security Infrastructure Supporting Access Network Operation....86
Figure A2-7: BRAIN Inter-Layer Interfaces....88
Figure A2-8: Interactions between Network Layer Component Problems....90
Figure A2-9: Radio Resource Management Message Exchanges....96
Figure A2-10: The No Coupling Architecture.....116
Figure A2-11: Call flow for SIP terminal mobility during HO between UMTS & BRAIN networks.....117
Figure A2-12: The Tight Coupling Architecture (a) Access via the GGSN; (b) Access via the SGSN.....118
Figure A2-13 Tight Integration at GGSN and HLR level....119
Figure A2-14: BRAIN Access to UMTS via the Iu interface (via SGSN)....119
Figure A2-15: Adapted UMTS Bearer Concept....120
Figure A2-16: BRAIN – UMTS Protocol Stacks (user plane)....120
Figure A2-17: Combined View of all BRAIN Public Access Alternatives....121
Figure A2-18: Basic AAA Components....127
Figure A2-19: AAA Model Adapted to BRAIN....129
Figure A2-20: AAA Brokers....132
Figure A2-21: Security Associations....133
Figure A2-22: GSM Handover....138
Figure A2-23: Diversity Combining....139
Figure A2-24: Soft Handover....139
Figure A2-25: Possible Diversity Combining in the BRAIN....141
Figure A2-26: Radio Resource Management Message Exchanges....144
Figure A2-27: Components of the Positioning Service....147
Figure A2-28: SLoP In the BRAIN Architecture....150

Figure A3-1: A Generalised Planned Handover....152
Figure A3-2: A Generalised Unplanned Handover....153
Figure A3-3: EMA-MIP Unplanned Handover....154
Figure A3-4: EMA-MIP Planned Handover....155
Figure A3-5: Framework for Smooth Handovers (unplanned MCHO)....156
Figure A3-6: Framework for Smooth Handovers (planned NCHO)....156
Figure A3-7: Fast Handovers in Mobile IPv6....157
Figure A3-8: FA Assisted Source-Triggered Handover....157
Figure A3-9: FA Assisted Target-Triggered Handover....158
Figure A3-10: Fast Handoffs in MIPv6....158
Figure A3-11: Fast Handovers for Mobile IPv6 (Network Controlled)....159
Figure A3-12: Fast Handovers for Mobile IPv6 (Mobile Controlled)....159
Figure A3-13: BRAIN Planned Handover....168
Figure A3-14: BRAIN Unplanned Handover....168
Figure A3-15: HAWAII Node State Diagram....171
Figure A3-16: CIP Node State Diagram....172
Figure A3-17: MIP Node State Diagram....173
Figure A3-18: HMIP Node State Diagram....174
Figure A3-19: Paging with BCMP....180
Figure A3-20: Hierarchical Path Updates....182
Figure A3-21: Meshed Path Updates....183
Figure A3-22: Hierarchy with Tunnels....184
Figure A3-23: Partial Mesh with Tunnels....185
Figure A3-24: BRAIN Network Topology....189
Figure A3-25: Example multi-BMG BAN....192
Figure A3-26: BAN with Attached MN....194
Figure A3-27: BMG/BAR Boundaries....195
Figure A3-28: Example Traffic Flow....195
Figure A3-29: Host Moves to BAR-3; case 1....196
Figure A3-30: Host Moves to BAR-3; case 2....196
Figure A3-31: Host Moves to BAR-3; case 3....197
Figure A3-32: Mobility Protocols Classification....208
Figure A3-33: Mobility Scenarios....209
Figure A3-34: Mobile Node Movement....210
Figure A3-35: BRAIN Proposals....211
Figure A3-36: The QoS-aware Nodes and QoS-signalling....215
Figure A3-37: Global vs. Regional Mobility Management....223
Figure A3-38: Classification of IP Mobility Proposals....225
Figure A3-39: Mobile Enhanced Routed TORA....227
Figure A4-1: Overview of RSVP usage within BRAIN....238
Figure A4-2: Regional Registration within the BAN....240
Figure A4-3: HMIP within the BAN....240
Figure A4-4: Aggregation of RSVP flows....241
Figure A4-5: (a) Cellular IP within the BAN, (b): HAWAII within the BAN....242
Figure A4-6: Multicast Scheme across the BAN....245
Figure A4-7: Simplified RSVP messages in a MER-TORA routed BAN....247
Figure A4-8: QoS signalling in the Network Nodes....253
Figure A4-9: QoS Signalling....255
Figure A4-10: Mobile Terminal States....259
Figure A4-11: Centralised BA architecture....263
Figure A4-12: Distributed BA Architecture....264
Figure A4-13: Concept of a Crossover Router....265
Figure A4-14: RSVP Path Repair....268
Figure A4-15: RSVP Messaging....268
Figure A4-16: QoS and Mobility....270
Figure A4-17: Session Establishment....275
Figure A4-18: Upstream Signalling....277
Figure A4-19: Downlink Signalling....277
Figure A5-1: Overall Architecture....287
Figure A5-2: ESI used by BRENTA Applications....290
Figure A5-3: Confirmed Service....291
Figure A5-4: Unconfirmed Service....291

Figure A5-5: Extended Confirmed Service 292
Figure A5-6: Extended Unconfirmed Service 292
Figure A5-7: Proxy Service 292
Figure A5-8: Notification Service 293
Figure A5-9: Service User / Service Provider 293
Figure A5-10: Enhanced Service Layer 294
Figure A5-11: Example's Network Topology 306
Figure A5-12 Usage of SetQoS Service primitives 307
Figure A5-13: Usage of AssignQoS Service primitives 309
Figure A5-14: Usage of QoSViolation 310
Figure A5-15: QoS aware Socket 311
Figure A5-16: Connection-oriented Service Provider 312
Figure A5-17: Connection-less Service Provider 313
Figure A5-18: RSVP QoS related Parameters and how they play together 316
Figure A5-19: Mapping of SetQoS Service 323
Figure A5-20: ChangeQoS Mapping 324
Figure A5-21: Mapping of QoS Parameter to Sender TSpec and Receiver FlowSpec 326
Figure A5-22: What is a QoS aware Application 327
Figure A5-23: Receiver QoS unaware Application 328
Figure A5-24: RegisterForFlow Service 329
Figure A5-25: Usage of RegisterForFlow and UnregisterForFlow 332
Figure A5-26: Register / UnregisterForQoSRequest 333
Figure A6-1: TCP/IP Protocols and IP_2W Interface 345
Figure A6-2: IP2W Convergence Interface 346
Figure A6-3: Neighbor Discovery Procedure 348
Figure A6-4: Address Acquisition Signalling 349
Figure A6-5: Centralised BA Architecture 354
Figure A6-6: Distributed BA Architecture 354
Figure A6-7: Primary Scheduling Model Choices 357
Figure A6-8: Planned MCHO Signalling 361
Figure A6-9: A Mobile-Controlled Planned Handover 362
Figure A6-10: Paging Areas in the BRAIN Access Network 363
Figure A6-11: Paging Request at the Mobile Nodes 364
Figure A6-12: Paging Request at the BRAIN Access Router 364
Figure A6-13: Paging Procedures 365
Figure A6-14: The Relation of Various Addresses 369
Figure A6-15: Handover Phases 378
Figure A7-1: Activity 2.4 Workflow 394
Figure A7-2: BRAIN Network Topology (D2.1) 399
Figure A7-3: Small Company-1 network topology 402
Figure A7-4: Small Company-2 network topology 402
Figure A7-5: Small Company-3 network topology 402
Figure A7-6: Mesh-1 Network Topology 403
Figure A7-7: University Campus-1 Network Topology 403
Figure A7-8: University Campus-2 Network Topology 403
Figure A7-9: University Ccampus-3 Network Topology 404
Figure A7-10: Train Station-1 Network Topology 404
Figure A7-11: Simulator Network topology 406
Figure A7-12: Planned Handover 406
Figure A7-13: Planned Handover Completed and Anchor Optimised. 407
Figure A7-14: Unplanned Handover 407
Figure A7-15: After second Unplanned Handover 408
Figure A7-16: Change of Anchor 408
Figure A7-17: TCP Throughput Sent and Received vs. Time 409
Figure A7-18: UDP Uplink & Downlink Throughput vs. Time 409
Figure A7-19: TCP (1024 pkt size) and UDP (CBR@64kbps & 48bytes pkt size) Packet Dropping vs. Time 410
Figure A7-20: TCP Transmitted Packet Sequence Number vs. Time 410
Figure A7-21: Received TCP Packets Sequence Number vs. Time 410
Figure A7-22: Uplink and Downlink Received UDP Packet Sequence Number vs. Time 411
Figure A7-23: Tahoe TCP Congestion Window vs. Time. 411
Figure A7-24: Simulator Network Topology 412

Figure A7-25: Planned Handover....413
Figure A7-26: Completed Planned Handover....413
Figure A7-27: Unplanned Handover....414
Figure A7-28: After Second Unplanned Handover....414
Figure A7-29: TCP Throughput Received Traffic vs. Time....415
Figure A7-30: UDP Uplink and Downlink Throughput vs. Time....415
Figure A7-31: TCP and UDP Packet loss vs. Time....416
Figure A7-32: TCP Maximum Transmitted Sequence Number....416
Figure A7-33: Tahoe TCP Congestion Window Size vs. Time....417
Figure A7-34: Network Scenario....418
Figure A7-35: VoIP Traffic....418
Figure A7-36: Simulation Network Topology....419
Figure A7-37: Handover....420
Figure A7-38: Snapshot Just after Handover....420
Figure A7-39: Throughput of VoIP Traffic when De-coupled....421
Figure A7-40: Packets of VoIP Traffic Lost per Second when De-coupled....422
Figure A7-41: Delay of VoIP Traffic when De-coupled....422
Figure A7-42: Throughput of VoIP traffic when Coupled....423
Figure A7-43: Packets of VoIP traffic Lost per Second when Coupled....423
Figure A7-44: Delay of VoIP Traffic when Coupled....424

List of Tables

Table A2-1: Summary of Coupling vs. Characteristics....122
Table A2-2: Security rules....135
Table A3-1: Basic Properties of the Handover Schemes....160
Table A3-2: Messages in the Handover Schemes mapped to the Handover Framework (Planned)....161
Table A3-3: Messages in the Handover Schemes mapped to the Handover Framework (Unplanned)....161
Table A3-4: Properties of Micro Mobility Protocols....209
Table A3-5: Forwarding Assurances for Flows....216
Table A3-6: HandoverTypes for Independent Flows....217
Table A3-7: Example Protocol for Each Category....226
Table A3-8: Summary of How Exemplar Protocols Tackle each Protocol Design Issue....228
Table A4-1: Parameter Information....273
Table A5-1: Service Primitives....291
Table A5-2: RSVP QoS related Parameters....316
Table A5-3: Mapping of SetQoS Service Primitives....322
Table A5-4: Mapping of SetQoSViolation Service Primitives....322
Table A5-5: Mapping of ChangeQoS Service Primitives....323
Table A5-6: Mapping of ChangeQoSViolation Service Primitives....323
Table A5-7: Mapping of ESI QoS Parameters to RSVP Sender TSpec....325
Table A5-8: Mapping of ESI QoS Parameters to RSVP Receiver TSpec and RSpec....325
Table A5-9: Mapping of ESI QoS Parameters to RSVP Receiver TSpec and RSpec....325
Table A5-10: Analyse the Sender/Receiver Scenario....327
Table A6-1: Router Advertisement Message....350
Table A6-2: Prefix Information Option....350
Table A6-3: Interface Identifier Based on EUI-64....350
Table A6-4: IEEE 802.1p Traffic Types....353
Table A6-5: Generic Link-layer Address Properties....370
Table A6-6: Ethernet Protocol Identifiers....370
Table A7-1: Small Company Scenario....400
Table A7-2: University Campus Scenario....400
Table A7-3: Train Station Scenario....401
Table A7-4: Global Network Scenario....401

List of Abbreviations

3G	Third Generation
4G	Fourth Generation
AAA	Authentication, Authorization and Accounting
ACAP	Application Configuration Access Protocol
ACK	Acknowledgement
ACTS	Advanced Communications Technologies and Services
AF	Assured Forwarding
AH	Authentication Header
AN	Access Network
ANG	Access Network Gateway
ANP	Anchor Point
AP	Access Point
API	Application Programming Interface
AR	Access Router
ARQ	Automated Repeat reQuest
ASP	Application Service Provider
ATM	Asynchronous Transfer Mode
BA	Bandwidth Allocator
BAN	BRAIN Access Network
BAR	BRAIN Access Router
BB	Bandwidth Broker
BBM	Break-Before-Make
BCMP	BRAIN Candidate Mobility Protocol
BCPN	Business Customer Premises Network
BD	Bounded Delay
BER	Bit Error Rate
BMG	BRAIN Mobility Gateway
BOF	Birds-of-feather (group)
BRAIN	Broadband Radio Access for IP Based Networks
BRAN	Broadband Radio Access Network
BRENTA	BRAIN End Terminal Architecture
CA	Certificate Authority
CBQ	Class-Based Queuing
CCoA	Collocated Care-of Address
CID	Cell ID
CIP	Cellular IP
CLI	Caller Line Identification
CN	Correspondent Node
CoA	Care of Address
COPS	Common Open Policy Service
CoS	Class of Service
CPN	Customer Premises Network
CPU	central processing unit
CRC	Cyclic Redundancy Check
CRP	Correspondent RSVP Proxy Server
CSCW	Computer Supported Collaborating Work
CSMA	Carrier Sense Multiple Access
DAB	Digital Audio Broadcasting
DAD	Duplicate Address Detection
DB	Data Base
DCPN	Domestic Customer Premises Network
DES	Data Encryption Standard
DHCP	Dynamic Host Configuration Protocol
DiffServ	Differentiated Services
DLC	Data Link Control
DNS	Domain Name System
DOI	Domain Of Interpretation
DoS	Denial Of Service
DS	Differentiated Services

DSCP	Differentiated Services Code Point
DSS	Digital Signature Standard
ECN	Explicit Congestion Notification
EDGE	Evolved Data for GSM Evolution
EF	Expatiated Forwarding
EMA	Edge Mobility Architecture
E-OTD	Enhanced Observed Time Difference
ESI	Extended Socket Interface
ESL	Enhanced Socket Layer
ESP	Encapsulating Security Payload
ETSI	European Telecommunications Standards Institute
EU	European Union
FA	Foreign Agent
FCC	Federal Communications Commission
FEC	Forward Error Correction
FIFO	First In First Out
FTP	File Transfer Protocol
GFA	Gateway foreign agent
GGSN	Gateway GPRS Support Node
GPRS	General Packet Radio Service
GPS	Global Positioning System
GRE	Generic Routing Encapsulation
GSM	Global System for Mobile communications
GTP	GPRS Tunnelling Protocol
HA	Home Agent
HAWAII	Handoff-Aware Wireless Access Internet Infrastructure
HIPERLAN	HIgh PErfomance Radio Local Area Network
HMIP	Hierarchical Mobile IP
HO	HandOver
HSCSD	High Speed Circuit Switched Data
HTTP	Hypertext Transfer protocol
IAB	Internet Architectures Board
ICI	Interface Control Information
ICMP	Internet Control Message Protocol
IDEA	International Data Encryption Algorithm
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IGMP	Internet Group Management Protocol
IKE	Internet Key Exchange
IP	Internet Protocol
IP2W	IP to Wireless Interface
IPSec	Internet Protocol Security
ISDN	Integrated Services Digital Network
ISL	IP Spatial Location
ISP	Internet Service Provider
ISSLL	Integrated Services over Specific Link Layers
IST	Information Society Technology
ITU	International Telecommunication Union
ITU-R	ITU Radiocommunication Standardization Sector
ITU-T	ITU Telecommunication Standardization Sector
IWU	Interworking Unit
L2	Layer 2
L2TP	Layer 2 Tunnelling Protocol
L3	Layer 3
LAN	Local Area Network
LMI	Local Management Interface
LOS	Line-Of-Sight (path)
MAC	Medium Access Control
MAN	Metropolitan Area Network
MANET	Mobile Adhoc Network
MBB	Make-Before-Break
MBS	Maximum Burst Size

MCPN	Mobile Customer Premises Network
MER	Mobile Enhanced Routing
MER-TORA	MER-Temporally Ordered Routing Algorithm
MH	Mobile Host
MIND	Mobile IP-based Network Developments
MIP	Mobile IP
MLD	Multicast Listener Discovery
MMP	Multicast for Mobility Protocol
MN	Mobile Node
MPEG	Motion Picture Expert Group
MSC	Mobile Switching Center
MT	Mobile Terminal
MTU	Maximum Transfer Unit
NAI	Network Access Identifier
NDIS	Network Device Interface Specification
NIC	Network Interface Card
NP	Network Provider
ns-2	Network Simulator version 2
OS	Operating System
OSI	Open Systems Interconnection
OSPF	Open Shortest Path First
PBX	Private Branch Exchange
PC	Personal Computer
PCM	Pulse Code Modulation
PCMCIA	Personal Computer Memory Card International Association
PcoA	Private Care of Address
PDA	Personal Digital Assistant
PDB	Per Domain Behaviour
PDU	Protocol Data Unit
PGP	Pretty Good Privacy
PHB	Per Hop Behaviour
PHS	Personal Handyphone System
PHY	Physical Layer. Layer 1 of the ISO/OSI reference model.
PILC	Performance Implications of Link Characteristics
PKI	Public Key Infrastructure
PLMN	Public Land Mobile Network
PN	Public Network
PPP	Point-to-Point Protocol
PPTP	Point-to-Point Tunnelling Protocol
PS	Packet Switched [domain]
PSP	Proxy Services Provider
PSTN	Public Switched Telephony Network
PWA	Personal Wireless Assistant
QoS	Quality of Service
RACE	Research and Development in Advanced Communications for Europe
RADIUS	Remote authentication dial-in user service
RAN	Radio Access Network
RED	Random Early Detection
RED	Random Early detect
RES	Radio Equipment and Systems, standardisation technical committee within ETSI
RF	Radio Frequency
RFC	Request for Comments
RIP	Routing Information Protocol
RM	Requester Module
RPF	RSVP proxy flag
RSA	Rivest-Shamir-Adelman
RSSI	Radio Signal Strength Indicator
RSVP	Resource Reservation Protocol
RTCP	Real-Time Transport Control Protocol
RTP	Real-Time Transport Protocol
SA	Security Association

SAP	Service Access Point
SBM	Subnet Bandwidth Manager
SDU	Service Data Unit
SeaMoby	Context and Micro-mobility Routing [Working Group]
SET	Secure Electronic Transmission
SGSN	Serving GPRS Support Node
SGW	Security Gateway
SIM	Subscriber Identity Module
SIP	Session Initiation Protocol
SLA	Service Level Agreement
SloP	Spatial Location Protocol
SLP	Service Location Protocol
SMS	Short Message Service
SNMP	Simple Network Management Task Force
SPD	Secure Policy Database
SSL	Secure Sockets Layer
TA	Terminal Adapter
TCP	Transmission Control Protocol
TE	Terminal Equipment
TM	Traffic Management
TOA	Time Of Arrival
TORA	Temporally-Ordered Routing Algorithm
TOS	Type of Service
TTFF	Time To First Fix
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunications System
USIM	UMTS Subscriber Identity Module
US-TDMA	USA Time Division Multiple Access (Cellular telephone system in the USA)
UTRA(N)	UMTS Terrestrial Radio Access (Network)
VBR	Variable Bit Rate
VCoA	Virtual Care of Address
VHE	Virtual Home Environment
VoD	Video-on-Demand
VoIP	Voice over IP
VPN	Virtual Private Network
VT	Virtual Terminal
VTS	Virtual Transport Service
WAN	Wide Area Network
WAP	Wireless Application Protocol
WCDMA	Wideband Code Division Multiple Access
WDLC	Wireless Data Link Control
WFQ	Weighted Fair Queuing
WLAN	Wireless LAN
WRR	Weighted Round Robin
WSA	Wireless Shopping Assistant
WWW	World Wide Web

1 Introduction

1.1 Introduction to the Deliverable

The objectives of the BRAIN (*B*roadband *R*adio *A*ccess for *I*P Based *N*etworks) IST project are [1.1]:

1. To facilitate the development of seamless access to existing and emerging IP-based broadband applications and services for mobile users in global markets.
2. To propose an open architecture for wireless broadband Internet access, which will allow an evolution from fixed Internet, emerging wireless/mobile Internet specifications and UMTS/GSM.
3. To facilitate new business opportunities for operators, service providers and content providers to offer high-speed (up to 20 Mbps) services complementary to existing mobile services.
4. To contribute actively to global standardisation bodies in the necessary timescales to impact significantly the international standardisation.

The BRAIN project is thus a wide-ranging research activity to develop an IP-based mobile wireless system complementary to current 2nd and 3rd generation mobile systems. It encompasses user applications and middleware [1.2], all the way through to the air interface, and in particular HIPERLAN/2 [1.3]. The network layer must support and unify the entire system; in this deliverable we (BRAIN Work Package 2) present the result of our work on the network layer.

Our overall vision is characterised by:

?? an innovative approach to mobile networks “beyond 3G”

?? an “all IP” access network, which should allow new services and business models as well as lower costs (what we mean by “all IP” is discussed in section 2.2)

?? the requirements for seamless handovers and end- to-end QoS

?? the desire for interoperation with non-BRAIN networks like UMTS

Areas of study have included: the design of the architecture for a BRAIN Access Network; protocol analysis and development; and simulation. The experience and knowledge gained are being made available to standardisation programmes in Europe and beyond. The follow up project, MIND [1.4], will build on the BRAIN project, including the development of a prototype system that demonstrates the practical feasibility of the BRAIN approach.

The overall project has followed a “top-down” approach, in other words driven by the requirements of end users. Those user requirements have been identified and described in [1.5]. Section 1.3 hints at this approach through outlining some of the ‘user requirements’ and the severe challenges they pose a network operator.

1.2 Structure and Scope of the Deliverable

An important part of the work described in this Deliverable has been to develop an architecture for a BRAIN Access Network. Whatever the fine details of the technical solutions we are proposing, they have one critical test to pass: how do they fit together, internally and with other elements of the environment (such as applications or the Internet) in order to support the activities of an ordinary end user of the new network. Thus the architecture needs to unify a disparate set of Internet protocols into a coherent mobile network. Section 2 describes and justifies the architecture and some of its implications. Included here are security (section 2.3.3) and inter-operation with non-BRAIN networks (section 2.4.2).

Other work has focussed on particular components inside the architecture, especially:

?? Mobility management – how to manage routing of IP packets around the network as terminals move (section 3)

?? Quality of service – how to ensure quality of service is provided to user sessions at the network level, including how to decide whether to admit a request (section 4)

?? The BRAIN protocol stack and its interfaces to applications/middleware and to wireless link layers (section 5)

The overall picture presented within this Deliverable has been validated through:

?? Simulations of specific aspects (within Sections 3 & 4). In particular simulations have validated the correctness of operation of the BRAIN Candidate Mobility Protocol (annex A7.4.1) and have examined the effect of coupling mobility and QoS (annex A7.4.3). Simulations are ongoing and will continue in MIND.

?? A "worked example" (section 6), which is a 'thought experiment' to show how the architecture and protocols presented in the other sections could be combined to solve end-user requirements (for instance those outlined in section 1.3). It is mainly intended as an aid to the reader of this deliverable but it is not intended to provide a set of BRAIN recommendations.

The structure of this Deliverable is a "core" report, plus a series of Annexes with detailed material. The Core summarises our key conclusions on the main subject areas, and briefly justifies them. The Annexes provide more extensive details for those interested in exploring a particular topic further. There are appropriate references from within the Core to the Annex material; note that Annex numbered 'X' corresponds with Core Section 'X' (BRAIN Interfaces are in different annexes).

1.3 Prelude – BRAIN Network Motivation, a User Perspective

In the BRAIN project proposal, Technical Annex 1 [1.1] introduced Carol, who was imagined as someone with a mobile terminal in a few years time. In this Section we re-visit some of Carol's requirements and comment on them from a **network** perspective. The rest of the Deliverable is really about how the BRAIN will enable all Carol's needs to be met by a future, "beyond 3G", BRAIN network operator.

> Carol is a demanding person. She has a mobile terminal and wants to be able to use it for all her applications. She wants:
>
> ??To have the same familiar applications whether she's on the move or at her desk. They should behave the same way for her wherever she is and whatever type of network she is using.
>
> ??To use applications that are real time and/or demand a high quality of service, for instance in terms of bandwidth, end-to-end delay and jitter.
>
> ??To get the latest applications. For example, a cool application that seeks out local screenings of her 'type' of movie; of course, it's an Internet application, and Carol wants to run it on her wireless terminal just like it was any ordinary computer.

Carol's application requirements suggest that the BRAIN Access Network (BAN) should deliver true end-to-end IP, so that all the IP services work as expected. Applications should be able to continue to do all the clever stuff they want, independent of the functioning of the network.

> Carol wants to be able to use her terminal all the time, everywhere - indeed, sometimes her life seems to revolve around it. She uses it at home, at work, around at friends', and in a variety of public environments. When Carol arrives at work (for example) her terminal discovers the network automatically, and decides that she should attach to it. She also expects the same privacy and security wherever she is that she would have from 'dedicated' devices like a home cordless phone.

The BRAIN Access Network has to determine that Carol is authorised to use the network, and configure the wireless link to prevent eavesdropping, and then Carol's terminal has to register with core network servers so that she can use applications. Almost all of this is actually being done with standard Internet protocols, for example AAA protocols to provide authentication and COPS to check policy. The BRAIN network layer has to provide Carol's terminal with an IP address and configure security.

> Carol makes an outgoing voice call to a colleague, Dorothy.

We assume that she is using a Voice over IP application, which carries out all the directory, session and gateway negotiations, and all this is transparent to the network layer. However, once the voice call starts, there are tight quality of service requirements on delay and jitter for the voice packets: these requirements have to be conveyed (somehow) to the network layer which in turn has to request the right services from the wireless link. The application and the network have to cooperate to adapt their behaviour to the characteristics of whatever the wireless link in use.

> Carol continues chatting to Dorothy as she moves around the company building.

The network must provide mobility support (not just portability) and deliver QoS, despite the wireless link. For example, her applications should continue to get high QoS even during a handover. This poses a lot of challenges to the terminal and the network, especially as the wireless link will often (normally) be the bottleneck. The network and her terminal have to cooperate to re-establish points of attachment and reconfigure wireless links, and then update the packet path through the access network so that downlink packets can still be delivered to the right place, and all this has to be done without any perceivable

interruption to the packet flow as seen by the voice application. All these are functions of the BRAIN network layer.

> Now Carol gets on the tram, whilst still chatting to Dotty.

Carol has now moved out of the range of the BRAIN Access Network and onto the public UMTS network. The challenge is to get the BRAIN and UMTS networks to inter-operate successfully, in terms of (re-)authentication and (re-)registration for example. The aim is to minimise the break in the voice call.

> Carol's call ends. Sometime later, her friend Enid calls her to discuss going out that evening.

From the network perspective, the additional difficulty is that Carol's terminal has gone into an 'idle' mode, so that the network doesn't know exactly where she is. This means it first has to search for and wake up the terminal; this process is called paging.

> Enid and Carol discuss what film they'll go to tonight, chatting over the terminal's video-cam and using the movie finder application to find out about local screenings and to watch video clips of trailers for the most promising options. Carol arrives home during their chat. Enid wants to see Carol's new cat. "Meow" says Pilchard. A film is chosen (Police Academy 13) and the evening is a success.

To the network layer, the movie finder is just a more complex version of the voice application: it requires the same quality of service for the session, but this time the network can make more decisions about how to multiplex the streams together to ensure that all of them get some comparable quality of service. There also needs to be an interaction with a location-based service, to identify the nearest cinemas. At home Carol's terminal automatically switches to a different air interface technology: one with a lower range, but higher bandwidth. For simple packet routing, the handover between the different technology types is handled almost the same as an ordinary handover, but the video-cam application can switch into its highest QoS mode network, and the access network may also re-optimise. Ideally this should happen automatically, so that Carol and Enid just notice some improvement in the video quality.

2 The BRAIN Network Architecture

The purpose of the networks activity within the BRAIN project is to extend the state of the art in IP to support mobile and wireless access technology, building on solutions already available from the fixed network. The requirements on the network include enabling the efficient use of wireless links and providing quality of service support for modern multimedia applications, as well as allowing interconnection with fixed networks. At this very high level, the scope of the networks activity is shown in Figure 2-1, which introduces three key components:

?? The terminal, or Mobile Node (MN). This is the actual mobile wireless device which is owned and used by end users, and is assumed to have an IP stack and applications, as well as one or more link layers, wireless and possibly wired. Note that there is no restriction to HIPERLAN/2 as a link layer; the BRAIN access network is intended to be applicable to any wireless link type.

?? The BRAIN Access Router (BAR). This is the last hop router in the wired IP network, and also contains the wireless base station functionality. Clearly, implementations which decouple the IP and wireless functions are possible, but any such substructure is not visible from the network layer perspective.

?? The BRAIN Mobility Gateway (BMG). This is the router at the 'other end' of the access network, and marks the interconnection point with the rest of the IP world. We require that any mobility specific routing functions are terminated at the BMG if they have not been terminated deeper within the access network.

It can be seen that the network layer plays a central unifying role, through which all the other parts of the system interact.

A key point here is that we have designed an *access* network, i.e. a network of limited scope: we cannot attempt to replace the current fixed Internet. So, the first key decision made is to define the scope of the access network, both logical and physical.
The second key point, where BRAIN differs from current mobile radio networks, is that the network is fully IP based – not simply using IP as a data transport, but using so far as possible the design principles which have made the Internet a success. But why is this a valuable exercise?

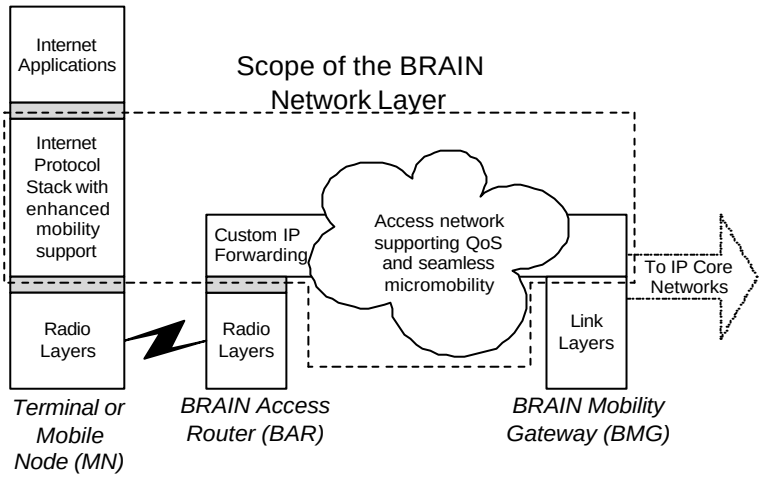


Figure 2-1: Scope of the BRAIN Network Layer

2.1 Why An IP Access Network

The main motivations designing a mobile access network based on IP technology fall into three categories: economic, engineering, and improvements for end users.

From the economic perspective, we have assumed that mobile network infrastructure should be based on the prevalent fixed networking standard, and there is no doubt that IP is the correct future proof choice. There are two major aspects to this. Firstly, IP will be ubiquitous, which is particularly significant for radio access providers for whom the cost of installing dedicated or specialised transport infrastructure can be prohibitive. Secondly, there are economies of scale, both in installation and operation: a single cabling and routing/switching network supports all mobile and fixed customers, public and private.

There are also sound engineering reasons for the choice of IP as a universal network layer. There is a growing consensus in the networking community that the philosophy embodied in the IP protocol suite has benefits over more traditional (connection oriented, cell or frame switching) networks. The main

aspects of this philosophy (and their corresponding advantages) can be summarised as keeping the network simple and stateless, with complexity pushed to the edge; and making the network modular, with open interfaces placed along natural functional boundaries.

Finally, we expect that in the future, all end user applications will actually be natively IP-based – that is, they will be written by people who take for granted the ability to send and receive IP packets. The movement towards 'pure' IP based access infrastructure means that applications available on fixed networks will inherently be available on mobile networks, and furthermore that they will behave consistently there, without their characteristics being submerged or modified by layers of mobile specific protocols. The same simplification will apply to terminals: a single IP stack (with all that that implies for simplicity of management and configuration) will be all that is necessary.

It is for all these reasons that BRAIN places an IP router at the very edge of the terrestrial network, a single radio hop from the mobile node. It should be noted that this is a major shift from current 2/3G mobile networks where the last hop router is actually the GGSN; and even wireless LAN systems still commonly attempt to handle the access point functionality as a layer 2 bridge. In contrast, our approach maximises the amount of fixed infrastructure that can be shared with other IP network users, and brings the engineering benefits noted above throughout the access network. Most importantly, it means that IP quality of service requirements from the network layer can be directly applied for every hop along the end-to-end path without intervening special-purpose protocol layers, which translates into more faithful and consistent treatment of application data flows

2.2 Design Principles – "What IP-Based Really Means"

To make progress in designing an IP-based access network, we need a set of design principles that capture the nature of IP networks, and can be used to guide us towards a design which solves a well defined problem and has a coherent internal structure. This is much more than simply saying 're-use existing IETF protocols wherever possible' – we need to select those protocols and make them fit together consistently. Instead, we have identified a set of fundamental design principles which embody the engineering goals of the project, and the most important of these are discussed here.

Our first rule is the end-to-end principle, which is one of the architectural principles of the Internet [2.1], [2.2]. The basic argument is that certain required end-to-end functions can only be correctly performed by the end systems themselves. In order to support this, the network should offer only some kind of minimal service to the end systems. In addition, providing specific functions within the network often also makes that network hard to evolve towards support for new services.

The end-to-end principle is sometimes reduced to the concept of the 'stupid network' [2.3]. In the mobile environment, the term is unfortunate since it is very hard for a high performance mobile access network to be truly stupid. Nevertheless, the underlying concept of minimal network functionality still applies (that is, the network should still look stupid). In the context of mobile access, this principle can be refined more concretely as follows:

?? Be independent of specific transport layers and applications, providing only an IP delivery service.

?? Be independent of what type of IP packets are being transported, and assume simply that packets are forwarded according to their IP header.

?? Minimise the number of special functions that are provided in the access network. Mobility support, especially for fast handover, can best be provided with network assistance, but this should (and can) be done in a way which does not reduce transparency.

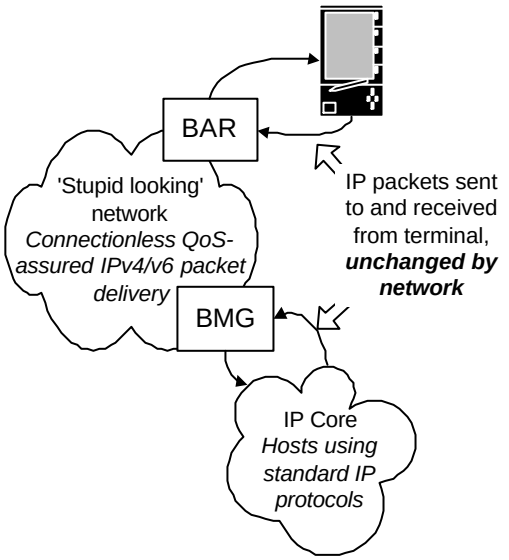


Figure 2-2: Network Transparency

Our second rule is that we should follow a layered design approach. Specifically, the access network should limit its functionality to providing IP packet forwarding, independent of applications. In mobile networks, it is common to end up with different layers of the protocol stacks being tightly integrated

together in the interests of efficiency. In the case of BRAIN, we have structured things in a more modular fashion with clear inter-layer interfaces. Specifically:

?? The network layer within the access network has a generic interface towards the link layer, such that new (and old) link layers can be exploited without wholesale network infrastructure redesign.

?? Where particular applications require optimised support, this is invoked and made available in a generic way, via a QoS aware service interface.

Note that some useful information or 'hints' about the link layer internals can, and even should, be made available for the upper layers to allow efficient operation above the link. However, this should be done in a way which does not change the upper layer protocol semantics, so the impact is limited to the implementation within a single network element – typically the terminal.

Our third rule is that the access network should be modular in its own design, and should fit into existing networks in a modular way. The purpose here is to minimise barriers to technology evolution and applies equally to upper layer services, link layers, and any components of the access network that lie between these. A related principle is that it should be easy to deploy a new system incrementally, for example, an initial system with limited performance followed later by performance extensions. As an example, we have chosen that the logical interface between the terminal and network which manages handover events should not enforce the use of a particular micro mobility protocol, but should allow network providers to choose appropriate solutions depending on their business model and deployment environment.

2.3 The Access Network Problem Definition

We start from a simple 'mission statement' to scope the problem:

The basic goal of the BRAIN Access Network is to make mobile wireless Internet access

look like 'normal' access through wired infrastructure.

In this context, 'normal' means firstly that other fixed networks and correspondent hosts must not be forced to make special adaptations because of the use of the mobile wireless technology. Functionally, the access network completely hides the mobility and wireless aspects, and these are only visible as performance impacts such as transient QoS variations - just as occur with other access systems. For applications on the mobile node itself, the sole effect is again that of QoS variations to which the application may or may not choose to adapt; the mobile and wireless aspects are again hidden within and below the network layer.

In conjunction with the design principles discussed earlier, this is enough to define how the access network looks to external entities.

2.3.1 Addressing and Routing

Fundamentally, a BRAIN Access Network (BAN) must allow a terminal to get an IP address to use in communicating with correspondent hosts in other networks; the BAN just routes packets to and from this address in a way which (externally) looks the same as any other IP network. Note that all addresses are locally assigned, which insulates the BAN from having to authenticate whether a MN is the authorised user of a particular address, and this means that relatively simple assignment protocols can be used, if necessary facilities provided by the link layer.

This approach, of routing to the mobile based purely on an assigned, local IP address, is shown in Figure 2-3. It can be seen that the entities within the BRAIN access network operate as pure IP routers (at least so far as packet forwarding is concerned), with no special treatment for encapsulation or decapsulation of 'home addresses' of the mobile node. Of course, this does not mean that Mobile IP is excluded, simply that its use is optional; the interaction with Mobile IP is described further below.

Although we have described the entities within the BAN as 'IP routers', this is mainly a reference to the fact that they forward packets simply based on their IP header. Simple prefix routing using a classic protocol such as RIP or OSPF to distribute topology information is not sufficient, since this would imply that the MN would have to change IP address every time it changed access router, and this address change would be visible outside the access network, violating our golden rule. Something new is required.

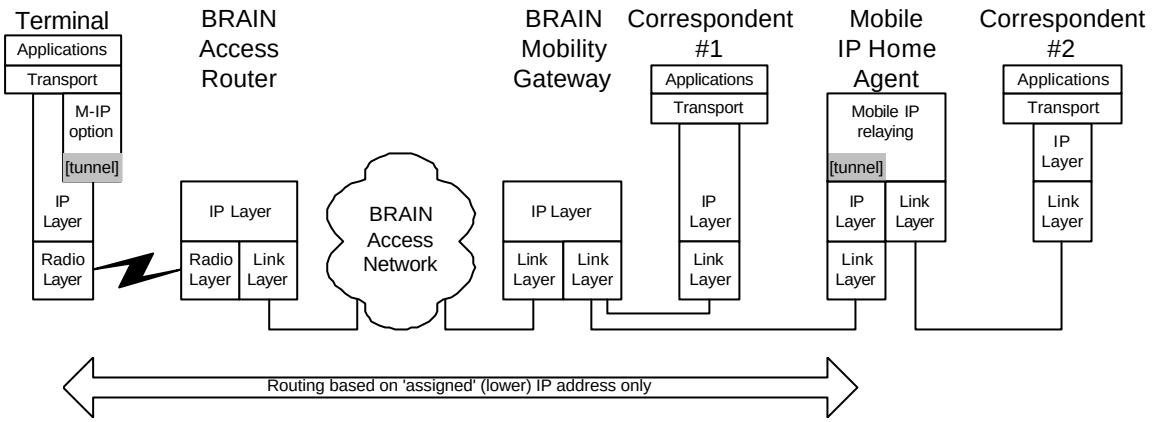


Figure 2-3: BRAIN Addressing Model

2.3.2 Scalability and Resilience

A second characteristic of mobile networks is the size – either geographical, or in number of users – that they may be required to scale up to, and with what reliability it has to operate. Building a small scale mobile network is actually quite simple; many additional problems arise once the requirements of a large public operator have to be taken into account. For example, resilience usually implies that we should strive hard to avoid single points of failure in our network design, but that then means that information needs to be replicated without impacts on performance. Given the rate of information update in mobile networks as users move around, this is a much harder problem than the resilient distribution of topology information that routing protocols have to achieve in today's fixed Internet.

A particular topological requirement for our BAN arises from the requirement for geographical scaling. Once an address has been assigned, the fundamental role of the BAN is to support seamless mobility of the terminal as it moves between access routers. In consequence, the allocated address must remain valid throughout the entire BAN, so there is a direct relationship between access network scalability and address allocation. There are essentially two options:

?? If seamless mobility within a single geographically limited area only is required, a BAN is allowed to interconnect with the core network at a single point, corresponding to a single BMG.

?? If seamless mobility over a very wide area is required, the performance of the Internet prevents us relying on BAN-BAN handovers to support this. Therefore, the combination of wide area support and seamless terminal mobility forces the use of multiple interconnects with the core.

This is one example of using the option for different protocols within the BAN depending on service provider requirements, since achieving very high scalability for a terminal mobility and QoS protocols is a hard problem and not relevant to (for example) a campus network operator. In either case, it is assumed that a BAN is under single administrative control, and seamless handovers between administrations are not catered for. The combination of these scenarios is shown in Figure 2-4.

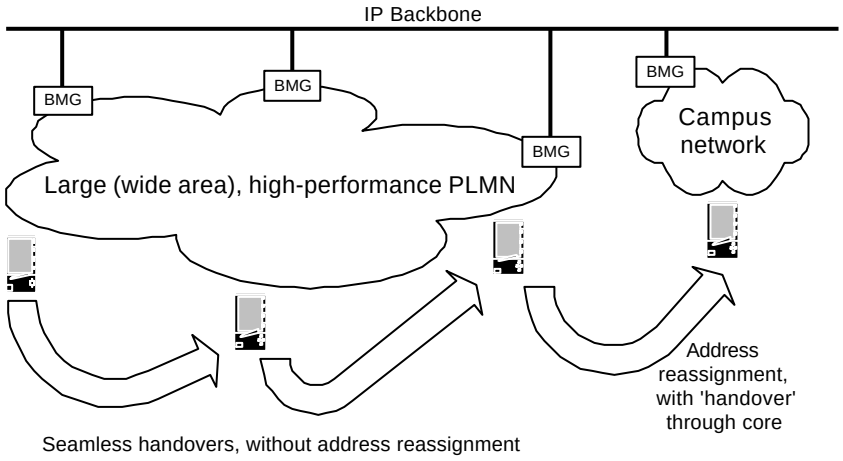


Figure 2-4: Network Scalability

2.3.3 Security

In line with our design goal of keeping the access 'minimally functional', there are two main security issues, which relate to the BRAIN access network. The first is that of Authentication, Authorisation and Accounting (AAA), which can be seen as the protection of the network from unauthorised users. The second is some basic requirement to provide confidentiality for user data over the air – real 'end-to-end' security, such as might be required for e-commerce applications, is assumed to be handled at the end system level, and cannot be the responsibility of the access network.

There is already a large and sophisticated set of standards to support AAA within the Internet, much of which has grown up around the need to support dial-up access [2.4]. These standards already support such advanced concepts as inter-ISP roaming, and are being further extended to such capabilities as hot billing and pre-pay. It is clear that these functions should be directly carried over into BRAIN– indeed, given the requirement to appear similar to fixed access, it is almost mandatory that BRAIN should maximise re-use of the corresponding protocols and standards. Details can be found in annex A2.4.

In the context of BRAIN, we can therefore summarise the AAA requirement as follows: A roaming user needs to be able to present credentials to the network which allow the user to be authenticated, and which allow the network to determine the resources to which the user is entitled. The architecture for supporting this is shown in outline in Figure 2-5, and is very simple: the MN has a logical security association with a 'local' AAA server AAAL, which also interacts with the BAR where issues such as handover authentication processing and resource request authorisation are concerned. Note that external security relationships are not shown explicitly, but could include for example a relationship between the MN and its home agent, or the relationships between the AAAL and other networks to support roaming. All these are supported by totally standard protocols.

The second security issue to be considered is confidentiality of the air interface. Here, we have identified two basic alternatives. The traditional approach would be to use a 'native' link layer encryption scheme; however, this usually has subtle interactions with whatever authentication mechanism is used and might require adaptations to the AAA protocol. More in keeping with our assumption of 'IP for universal access' would be to use IPSec between the MN and some fixed point within the access network; however, this is definitely a heavyweight solution, and there are definite impacts on air interface efficiency. Construction of generic yet efficient and wireless-friendly security solutions remains a challenge, and largely one for future study.

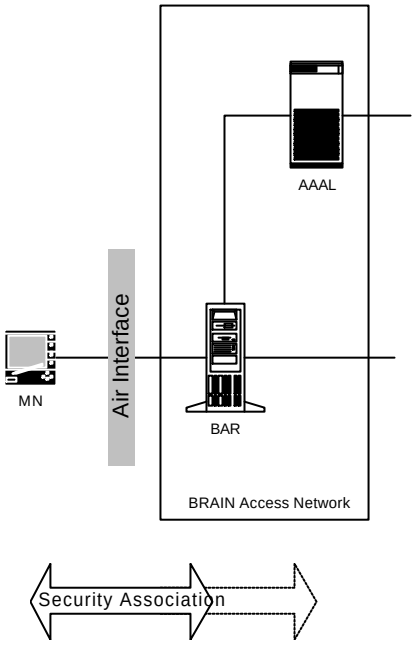


Figure 2-5: Security Associations

2.3.4 Radio Resource Management

As well as the problems of addressing, scaling and security, the network layer for any wireless access network must clearly have some interaction with the radio resource itself. Handover will almost invariably be triggered by radio conditions, and a decision to admit a session can only be made taking into account the detailed flow characteristics of all the traffic in that cell. Thus, the two central problems of the BRAIN access network - mobility and QoS - are intimately linked with the problem of radio resource management.

With the aim of being independent of any specific air interface, we have chosen in our work not to study radio resource algorithms in themselves. Instead, we have adopted a generic approach, whereby the radio specific aspects are handled using message flows whose content is opaque to the network layer. Only the logical distribution of these messages to RRM algorithms, and the triggers that these algorithms generate (such as 'recommend handover to cell X') are considered. More explanation of this architecture is contained in annex A2.6 and is intimately related to the functionality of the IP2W interface described in section 5.3.

2.4 External Interactions

2.4.1 Global IP Mobility

The Internet is already rich in protocols which support some kind of 'global mobility', allowing users access to the services of their home networks wherever they may be. We will consider here two examples. Firstly, we have the classic Mobile IP protocol, which supports global terminal mobility, the ability of a terminal to change its point of attachment – in particular, between access networks of different administrations – without session interruption. Secondly, we consider SIP, a signalling protocol used to build multimedia applications, which provides global personal mobility, the ability of an end user to use services from any terminal at any location. How do these protocols work in the context of the BRAIN network architecture?

In line with our guiding design principle of network transparency, our decision has been to support these functions without modifying the basic BRAIN network function of 'transport the IP packet unchanged'. Supporting SIP mobility in this way is actually very simple. The MN just acquires an IP address on attachment to the BAN in the standard way, and this address is then used during registration with the user's chosen SIP server, which could be anywhere in the global network. From this stage, the session negotiation messages and data packets are seen by the BRAIN network simply as IP packets, and no special treatment for them is required.

Support for Mobile IP is a little more subtle. As before, the MN acquires an IP address using the access network procedures, and is then able to use this as a Collocated Care-of Address (CCoA) in the MIP sense. The subsequent registration messages with the Home Agent (which again can be anywhere in the Internet) or other Binding Updates (e.g. for route optimisation) are totally transparent, as is the actual user traffic, whichever Mobile IP encapsulation or routing method is used. No Foreign Agent is involved. Note that in the meantime, the MN can also use the same (CCoA) address for SIP applications or anything else, such a Virtual Private Networking applications (e.g. based on L2TP or IPSec). Indeed, the MN could also use SIP and MIP in conjunction, by registering its home address with the SIP server; all of this is at the discretion of the user and MN, and is invisible to BRAIN.

We must stress that this complete decoupling of local and global mobility is quite different from many current investigations in the IP world, which attempt to solve all mobility problems with a set of extensions closely coupled to (some version of) Mobile IP. We have preferred to support local mobility in conjunction with any global mobility approach or combination of them, and to enable the solutions to these two quite different problems (intra- vs. inter-domain) to evolve independently. The price we have paid is that some optimisations of Mobile IPv4, specifically the use of Foreign Agent care of addresses, are now much harder to support. On the other hand, the CCoA approach is the only option in MIPv6 and is preferred in MIPv4 except for address shortage reasons. In the timescale anticipated for BRAIN in the real world, this approach seems reasonable. Indeed, it is consistent with our view that in the future, IP will become the universal protocol, which should be supported natively by all access technologies and on which all other protocols will be built.

2.4.2 UMTS and Other Mobile Networks

At least in the initial phases, BRAIN networks cannot stand alone. To achieve our fundamental goal of universal access to the services and applications discussed, it must be possible to use BRAIN in conjunction with other public mobile networks. In the timescales that we envisage, in practice this means we must be able to operate BRAIN and UMTS networks together so that their capabilities complement each other. And yet, it will be clear that the architectures underlying the two network types are fundamentally different in many respects – so how can this interoperation be realised?

In an initial analysis, concentrating on the known cases of UMTS R'99 and R4, we have identified three broad classes of interworking approach. These classes reflect different tradeoffs between on the one hand the required degree of modifications to standards and ease of development, and on the other hand the seamlessness of the interworking and amount of infrastructure commonality. Detailed analysis can be found in annex A2.3.

The simplest case can be called the 'no coupling' approach, where the BRAIN and UMTS networks are used simply as completely independent packet switched access networks. Users would have separate contracts for each network, and application or other servers (e.g. to support SIP or Mobile IP) might be supported by third parties, independent of either access provider. This approach clearly allows very rapid introduction, but at the cost of rather limited integration of the applications and infrastructure between the two environments. Handover performance for hard switching between the two will be poor, although it might be possible for a single terminal to maintain parallel connections on each network to mitigate this.

The next level up is referred to as 'loose coupling'. Here, the BRAIN access network is treated as a peer at the same level as the UMTS packet switched (PS) domain. Authentication message flows and other packet domain signalling flows (typically SIP in the UMTS 'All-IP' model) are directed towards the same core network entities. This approach supports a single operator offering a common service portfolio via either access type, with the fact that there are two networks largely hidden from the user. The main drawback is that handover is still not seamless; in particular, BRAIN-UMTS hard handover will necessarily force a change of local IP address.

The closest level of integration is referred to as tight coupling. Here, the BRAIN network becomes part of the UMTS network, attaching either to the GGSN as an extension to the PS domain, or to the SGSN as a new type of RAN. Here, the level of integration is very close, and mobility support within the packet domain will be good provided the mobile always remains attached to the same GGSN. However, the amount of modification to the UMTS or BRAIN interfaces is much greater, and in the case of attachment across Iu to the SGSN may not be practical at all.

Ultimately, the choice between these approaches will be driven largely by commercial and deployment issues, and cannot be predicted at this time. However, we have at least given an introduction to the options and the issues that they raise. Please note that this discussion parallels the equivalent investigations being undertaken in ETSI BRAN for the particular case of HIPERLAN/2, and the terminology and scenarios there are aligned at a high level with what is used in BRAIN. Future developments in ETSI BRAN and UMTS standardisation may point a clearer path for the evolution towards the true all-IP mobile access network solution.

2.5 Access Network Components

We have described what the BRAIN access network is, what it has to do, and how it fits into the rest of the IP and mobile world. We believe we developed an access network design that goes a long way towards meeting these goals; it just remains in the rest of this report to describe the internal components that make it work. There are three main groups of these.

The first is related to mobility management, which includes the problems of local seamless handover, idle mode and paging, and of course the routing capabilities within the access network that allow these to take place without constant address reassignment. All of these, especially the last, have major impacts on the internal architecture of the access network, in terms of location of functionality in different components. The BRAIN solution for mobility management is described in section 3, which includes a candidate overall protocol which integrates the solutions for handover, idle mode and routing. This has provided a concrete architecture to enable further analysis, and has been successfully simulated to verify correctness.

The second major group of components relates to quality of service. The focus here has been to identify how the specific problems of mobility and radio access impact on the provision of quality of service end-to-end, and what requirements on external networks might be needed to enable quality of service mechanisms within the access network to operate. Given the rapid evolution of QoS concepts even in the fixed Internet, it is an exceptionally difficult problem to work out how to adapt them to the mobile wireless environment; nevertheless, section 4 provides a standards-based baseline QoS architecture which addresses many of the issues, and also describes a set of extensions which can be used to optimise the QoS solution for particular scenarios.

The final area that has been considered is that of inter-layer interfaces, specifically above the TCP/IP stack (towards applications) and below (towards wireless link layers). Any attempt to solve the network layer problems in isolation runs the risk of providing a design which ignores critical application requirements, or cannot be implemented over real link layers. The service interfaces described in section 5 define a contract between the network layer and the rest of BRAIN, and have served as a guarantee that we do not fall into this trap.

3 IP Mobility Management

3.1 Introduction

The interest of this Section is terminal mobility (also known as host mobility) in a BRAIN Access Network (BAN). Internet routing protocols have traditionally been designed with the assumption that terminals are fixed, i.e. packets are routed to a particular point of attachment. In order to support mobile terminals, new protocols and modified architectures are needed - there have been a vast number of proposals in the literature and at the IETF. In the BRAIN project, we have performed a critical analysis of them through an Evaluation Framework [3.1], extracted the key functionalities that must be realised [3.2], and have also developed a "BRAIN Candidate Mobility Protocol" that may be suitable in many scenarios. The Section is brief; in-depth material can be found in annex A3.

The key objective that terminal mobility in the BAN must do is to ensure that packets are delivered to a MN after it has moved onto a new BAR. Amongst the characteristics that we would like the solution to have, are that it:

?? Minimises mobility signalling traffic, both globally and within the BAN
?? Provides seamless handovers, i.e. without significant delays and without loss of packets
?? Scales well, so it can be used when there are many MNs
?? Is robust, i.e. supports multiple routes or rapid re-routing
?? Is compatible with other Internet protocols

We have concluded that there are three main functions that an IP-mobility solution must provide. Since each concentrates on meeting different requirements, they can be analysed largely separately:

1) Handover Management (section 3.2)
This refers to the impact of handovers on the MN. It deals with the local signalling involving the MN and BARs to facilitate reattachment to a new BAR. The aims are to:

?? Minimise packet loss and delay during a handover
?? Make use of any "triggers" available (e.g. information that a handover is imminent from the MN at the link layer or from the network), in order that action can be taken in advance of the actual handover
?? Allow the possibility of passing context (QoS, security, header compression state, etc) from the old to the new AR, and also any buffered packets
?? Ensure that a planned handover can fall back gracefully to an unplanned one (in case it fails), and that the same actions can happen (transferring buffered packets and context)
?? Allow inter-technology handovers (if the MN can support them)

2) Path Updates (section 3.3)
This refers to the mechanism for installing information in the interior of the BAN so that packets can be successfully delivered to the MN at its new BAR. Its requirements include:

?? Be scalable – to cope with a big network with lots of mobiles [3.3].
?? Be robust – there should be no single point of failure and there needs to be quick automatic recovery from network node and link failures.

The Path Update solution should allow multiple gateways (BMGs), because this will help scalability and reliability (section 3.4).

3} Support for Idle Mobile Hosts (section 3.5)

For an idle MN (i.e. one not actively transferring packets), a combination of paging and location updates reduces signalling messages over the air and in the BAN, and saves router state and terminal power. In other words the aim is to reduce the overhead from (respectively) the Handover Management and Path Updates. Paging needs to be scalable and reliable.

3.1.1 Overall Approach and Interactions of Key Functions

In order to separate handover management and path updates, which are traditionally closely coupled, we consider the path update processing to be an activity which takes place after the handover is complete. Therefore, we have defined a handover management framework, which sets out the semantics of local signalling between the MN and the access routers. One key idea is that during a planned handover, signalling only involves the MN and the old and new BARs (i.e. not other routers within the BAN). It is

only when the mobile has finally moved onto the new BAR that a Path Update message is sent. The principle is to separate the concerns of (vertical) path updating in the mobility protocols from the (horizontal) handover signalling. This separation means, for example, path updates don't have to ensure a seamless handover – that is a requirement that handover management deals with – and conversely, handover management doesn't for example have to worry (much) about scalability.

The handover management framework must clearly define its interface with path updates. The separation of handover from path updates is a critical step in trying to specify a standardised protocol between the MN and the access network. This would allow deploying and developing various micro-mobility schemes without a need to change the operation of MN.

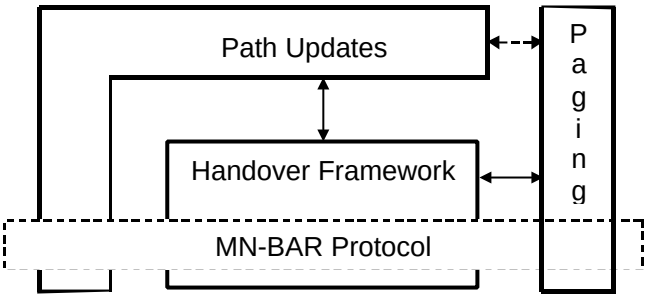


Figure 3-1: Relationship of Main Functions of IP Mobility Solution

3.2 Handover Management

3.2.1 Handover Problem

A handover may incur disruption of service as perceived by the MN user. This may be due to latency in diverting the routing path to the new access router, or dropped packets. Therefore, mechanisms are needed to ensure that packets are not lost and that they reach the MN without excessive delay.

In basic Mobile IP, the MN discovers new access routers by movement detection that relies on periodical network-layer advertisements and on other Neighbour Unreachability Detection mechanisms. Moreover, Mobile IP only specifies an unplanned (reactive) handover where the previous router may temporarily tunnel packets to the MN's new point of attachment. These techniques alone are inadequate for achieving seamless handovers. Therefore, if possible, handover preparations should be started while the MN is still connected to the old access router. Such planned (proactive) operation would also allow for interaction with network resource management, for example in the form of context transfer negotiations to ensure service availability. However, because handover may be time-critical due to environmental conditions (e.g., abrupt degradation of radio signal quality), handover signalling should not assume prolonged connectivity to the old access router.

Besides delays induced by detection of movement, other delays may be caused by address acquisition and authentication that may be necessary when the MN moves across borders of administrative domains. We don't consider these issues here, because the focus is on intra-domain handovers, whereby address acquisition can be avoided or expedited and the MN's session keys and other context can be distributed among the access network nodes.

3.2.2 Existing Solutions

As a baseline, Mobile IP handover scheme includes packet forwarding from previous Foreign Agent or router. This mechanism can be conceived as a prototype for a general unplanned handover protocol.

The overall goal of traditional micro-mobility designs has been to achieve seamless handovers, and to minimise mobility signalling traffic both globally and within an access network. These efforts have resulted in suggestions for regional mobility protocols that achieve fast convergence of the routing path by localising the path update signalling in a tree structure. In a large network, this necessitates multiple levels of routers maintaining host-specific routes, which clearly limits their applicability.

Recently, "complete" micro-mobility proposals have been broken up into separate specifications that tackle handover-specific sub-issues. The basic idea is to perform time-critical data path diversion locally between the access routers, which relaxes the need for using multiple-level hierarchical network topologies to achieve fast path updating.

It has been also recognised that seamless handovers cannot be achieved by using network-layer movement detection mechanisms. Therefore, virtually all proposals suggest some link-layer support that triggers the handover execution. This triggering may happen either at the MN or at the old access router (in unplanned handovers it may happen at the new access router). So far, the required convergence layer functionality has been out of the scope of IETF work but this has been considered by the BRAIN IP2W interface (see section 5).

A review of the current IETF handover proposals can be found in annex A3.1.

3.2.3 Conclusions on Handover Management

In this section, we define the scope of our handover procedure and present a layout of the proposed handover scheme. The design is based on an analysis of functional requirements for the handover signalling, which have been refined after analysis of existing IETF handover protocol proposals (for more information, see annex A3.1). A basic requirement has been that the protocol should be adaptable to various path update schemes (which could be derived from existing micro-mobility protocols that have been adapted to conform to BRAIN design principles). A particular adaptation is presented in section 3.6.

A protocol for intra-domain handovers specifies the signalling between the MN and the old and new access routers (OAR and NAR). As a conclusion from the handover protocol design we can summarise the required primary characteristics. The handover:

?? is planned, but can fall back to an unplanned handover. Only proactive operation ensures that the MN's service requirements can be fulfilled during and after a handover. This can be achieved by contracting with one or more candidate access routers. Also, packet loss can be avoided by buffering and bi-casting techniques even if link-layer connection set up is slow. A planned handover may not be possible or it may fail, for example, due to sudden loss of radio connectivity to OAR. Therefore, a graceful transition to an unplanned handover phase must be available.
?? is "mobile-controlled" (i.e., the network may assist in the handover or constrain it but the MN has the final control of the handover). The main reason is that the MN is best placed to understand the application's and the user's transitory needs, which cannot be satisfied by fixed policies.
?? does not assume any special support from the link-layer (e.g., make-before-break connection) but can make use of special features. For example, make-before-break is possible if the MN supports several access technologies. Furthermore, the link layer may be able to give indications of handover-related events, which should be used to expedite handover initiation and execution.
?? assumes a "semi-static" IP address for the MN within an administrative domain. This means that the MN's routable IP address does not change at an intra-domain handover, which is a direct consequence of BRAIN's design principles.

Thus, we do not assume that the MN can communicate simultaneously with OAR and CARs (Candidate Access Routers). However, we make the assumption that the MN is able to listen to CARs' broadcast advertisements while still being connected to OAR. That is, we don't require multi-homed MNs but we assume that the MN can temporarily tune its receiver to neighbouring channels.
The observations about the assumed link-layer support have been be fed as input to the BRAIN "IP to Wireless Convergence Interface" (IP2W) specification (see section 5).

The protocol provides signalling mechanisms that allow defining:

?? how OAR or the MN knows that a handover is needed,
?? how the MN and/or OAR determine CARs and inform each other about them,
?? how the MN's identification and other context are conveyed to CARs,
?? how OAR or a CAR informs the MN that that the handover is possible or has succeeded,
?? how IP packets are forwarded from OAR during a handover (to avoid packet loss), and
?? how and when path updating is performed.

A planned handover is performed when the MN is able to make preparations for the handover while still being connected to the OAR. If the MN abruptly loses its connection to OAR, or the planned handover fails for other reasons, the MN may fall back to the unplanned handover. The following sections briefly describe the planned and unplanned variation of the handover protocol at a high level (see Annex A3.1.5 for a more detailed description).

3.2.3.1 Planned Intra-domain Handover

The basic protocol elements involved in a planned handover are illustrated in Figure 3-2. The protocol operates as follows:

First, a link-layer trigger either at the MN or at the OAR indicates a need for a handover. Then, the MN and OAR agree on the CARs, and the MN requests a handover to one or more of the candidates. OAR requests for the handover willingness from the CARs by sending the MN's context to them. OAR informs to the MN of the CARs that have acknowledged the handover request. OAR also starts tunnelling packets to CARs. Finally, the MN registers with a NAR, which updates the path and confirms the MN's registration. After the downstream routing path has been diverted to NAR, packet forwarding at OAR can be terminated. This can be achieved by a timeout mechanism at OAR or sending an explicit signal from a "cross-over" router or NAR (not shown in the figure).

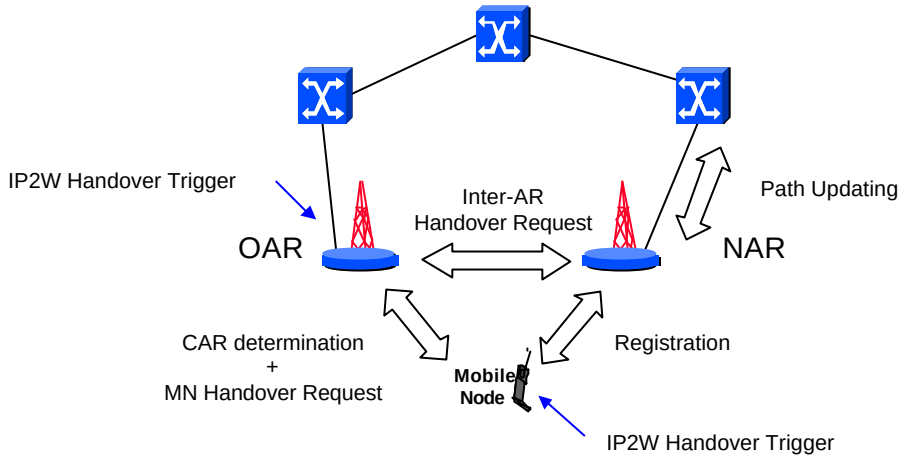


Figure 3-2: BRAIN Planned Handover

3.2.3.2 Unplanned Intra-domain Handover

The course of actions in an unplanned intra-domain handover is illustrated in Figure 3-3:

First, a link-layer trigger (e.g., loss of connection), or another indication that signifies a need for a handover, occurs at the MN. The MN establishes a link with NAR and registers with it by including the identification of OAR in the registration request. NAR solicits for the MN's context information from OAR. OAR responds by sending the required context information and starts forwarding downstream traffic to the MN by establishing a tunnel to NAR. Finally, NAR updates the routing path and acknowledges the MN's registration. (This last step could be done in parallel with the inter-BAR process, provided that the NAR already has a security context for the MN.)

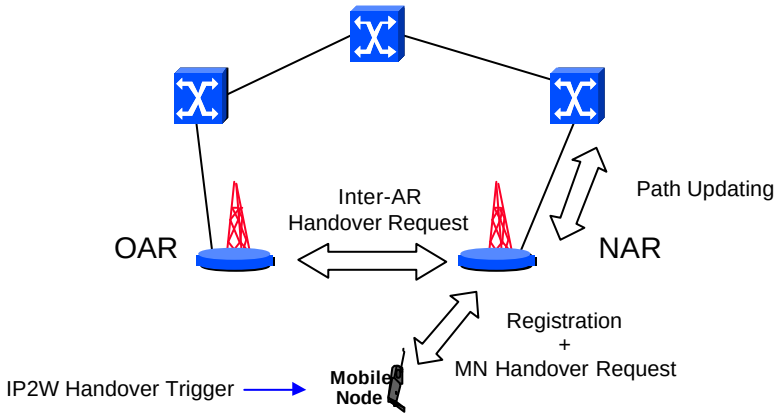


Figure 3-3: BRAIN Unplanned Handover

3.3 Path Updates

3.3.1 The Path Update Problem

There are two distinct cases of packet forwarding within the BAN. The first of these is the 'traditional' routing problem, making sure that nodes within the fixed BAN infrastructure are reachable. The other case manages packet forwarding to mobile nodes (MNs) attached to the BAN through a BAR. Because the point of attachment of a MN can change dynamically, the network needs to track the location of the mobile in order to forward packets to it.

The specific role of the path update mechanism is to install host-specific state in the network to allow the packet forwarding nodes to adapt to the movement of MNs. Unlike conventional routing where IP addresses are stable and have implicit geographical significance, routing to mobile nodes conceptually requires location (and routing) information to be maintained for each node. Whenever a mobile node moves, path information must be checked or updated to ensure that it remains reachable. Routing in the fixed Internet is far from trivial. With a large and highly mobile user community, routing becomes a complex problem.

For scalability and performance reasons, it is clear that limiting the per-host state in the network is necessary. It is also clear that it is insufficient to install a single route to a given MN using standard routing protocols. (For the packet forwarding scheme to take advantage of route diversity, the routes must be known). The challenge for path updates is to find an efficient method of managing this state information. For more details, refer to the annex A3.1.5.2.

3.3.2 Existing Solutions

Early analysis of protocols identified a number of different classes. The path update mechanisms can be generalised and abstractly considered as falling between two extremes. Examples of these degenerate cases are mobile-IP (one location aware node, the Home Agent) and HAWAII (many nodes participate in a mobile aware path-update process). Within [3.2], these approaches are referred to as 'gateway-centric' and 'hop-by-hop', respectively. The protocols can also be classified according to whether they are 'proactive' or 'reactive' in nature, i.e. whether the location information is updated on every MN movement, or the MNs are searched for on demand, when there is incoming data. Figure 3-4 illustrates the classification of some Path Update protocols, and how they could be converged to three categories.

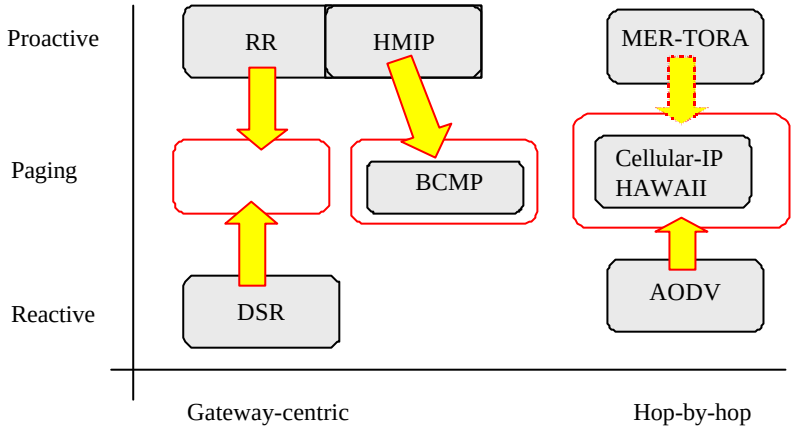


Figure 3-4: Mobility protocol convergence

The remainder of this section discusses how some current mobility management protocols handle path updates. Further details are in [3.1] and annex A3.3. Weaknesses with these approaches are explored as a way of identifying what issues a BRAIN operator should consider.

Some solutions simplify the problem by making assumptions about topology – most commonly a tree-like form. Examples are Cellular-IP and HAWAII. In these cases there is always a well-defined 'cross-over router' in the uplink path. Optimal paths can be installed and maintained by sending packets towards the root of the tree. Failings of such schemes are that there is a single path to the mobile and the root is a single point of failure.

If the 'tree-like' topological restriction is lifted to allow a partially meshed structure the path update and packet forwarding issues become significantly more complex. For example, per-host information is frequently distributed between a number of nodes. This information needs to be updated in a synchronised manner and accurately maintained to avoid introducing routing loops.

The tunnel-based approaches such as Mobile IP are topologically independent, but have other flaws. Basic Mobile-IP has a single point of failure. Enhancements such as Hierarchical Mobile IP construct a logical tree that suffers from the same disadvantages as any other tree-based approach. Tunnelling complicates for instance the provision of QoS, although the extent to which it does is highly contentious (see annex A3.3.3).

MER-TORA uses an adapted *ad-hoc* networking protocol to solve both the fixed and mobile routing problems. The problem with an approach such as this is that it requires all routers to be upgraded with the

new protocol and (perhaps) the complexity of the protocol. Although it supports arbitrary networks of nodes, some topologies are managed more efficiently than are others.

Per host schemes (e.g. HAWAII, MER-TORA) must control the signalling load from path updates. This can be helped by address aggregation: for example, at the start of a session a MN obtains an IP address 'owned' by the current BAR, so that fully prefix-based routing can be used. To limit the signalling as the MN moves, the MN must not be allowed to 'hoard' its IP address for too long.

Having identified that no existing solution maps neatly to all the BRAIN requirements, the next step is to identify what comprises a good answer.

3.3.3 Attributes of Solutions

A path update solution can be examined against a number of criteria. The aim is to identify those attributes that are considered desirable, based on our analysis above. The points below are suggested as a way of identifying how effective a given scheme is as a solution.

Within the BAN a path update scheme should:

?? **support address aggregation:** this is a general efficiency issue, rather than a direct function of path updates, but it must be supported.

?? **be independent of network topology:** unlike many existing architectures, the solution should allow deployment in networks with arbitrary connectivity. This is particularly helpful for evolving legacy networks.

?? **avoid routing loops:** more generally, path updates must complete within a finite time (convergence) and leave a stable forwarding path. Any solution that allows packets to be lost in routing loops is unusable.

?? **support multiple paths:** for resilience, scalability and flexibility, packets flowing between a CN and MN across the BAN should not be limited to a single path.

?? **require limited signalling overhead:** where two otherwise similar solutions exist, one that places a lower signalling load on the network is preferred. Failures within the network should also be managed without excessive signalling load.

?? **be robust in the presence of link and node failures:** the failure of any link or node (other than the current BAR) in the network should not prevent packets from reaching the MN (or the CN).

?? **be simple:** hard to quantify, but a simple protocol is easier to test and maintain. There are many components to a mobility management solution, and each part should contribute as little as possible to the overall complexity.

?? **allow separation of local from global mobility:** this is a requirement from the Architecture section (section 2)

These may be considered as idealised requirements or design goals for a protocol. More details can be found in annex A3.3.5.

3.3.4 Conclusion

This section has given an overview of the path-update solution for the BRAIN mobility management solution. For more details, refer to the annex A3.3. This work has increased understanding about how path updates fit into mobility management.

3.4 Scalability and Resilience

It is intended that a BRAIN Access Network be highly resilient and scalable to arbitrary size. This is an important area of study for BRAIN, and highlights an aspect of improvement over many existing solutions.

A BRAIN Mobility Gateway (BMG) is defined as a special purpose IP router hiding any BRAIN-specific routing functionality. That is, its goal is to make the BAN look like a normal, fixed network to any entity outside of the BAN. The support of multiple gateways can be seen to be essential for realising BANs. However, this area has received little coverage in existing proposals (see annex A3.4).

This section discusses the motivation for requiring the BAN to support multiple BMGs and the issues that arise from this.

3.4.1 The Case for Multiple BMGs

There are three clear reasons for using multiple gateways. These are presented along with suggestions of alternative solutions that do not require the use of multiple BMGs.

?? **Throughput Scalability:** Having a single point within a BAN through which all up- and down-link traffic flows is a bottleneck. If the BAN supports multiple BMGs, then this provides a very natural way to provide scalability. This problem could also be mitigated with highly scalable BMGs, although this would add architectural complexity to the design of the BMG.

?? **Geographic Scalability**: Imagine a country-wide BAN where all traffic to a mobile node has to flow through a single BMG. This would seriously compromise routing efficiency. For a small BAN, this routing sub-optimality may not be serious, but for large BANs a solution is important (see also section 2.3.2 for a discussion of address validity). Realistically, this requires multiple BMGs.

?? **Resilience:** Resilience is the most obvious reason for wishing to support multiple BMGs – avoiding a single point of failure (see also section 2.3.2). It is also the hardest problem to solve, particularly in a way that hides failure from the end-user. An alternative to using multiple gateways is to design the BMG so that it has high availability.

?? **Overall:** Although it is assumed that the gateway is highly reliable, the combination of the above factors makes support for multiple BMGs extremely attractive. This is because in solving how to manage multiple gateways within a BAN, all the benefits outlined above are automatically available. Moreover, networks can be scaled (or gain additional robustness) just by adding extra gateways.

3.4.2 Routing

Consider the following simplified BAN structure, with multiple BMGs:

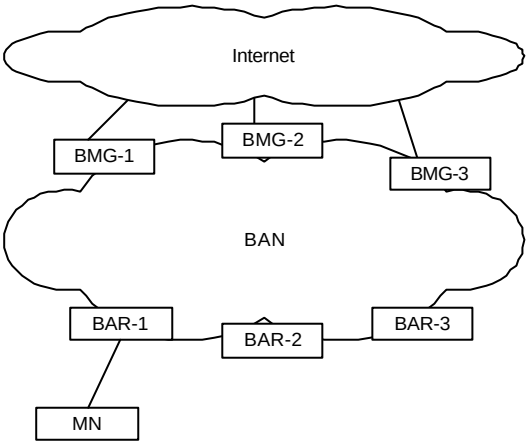


Figure 3-5: A BAN structure with multiple BMGs

In that the BMG makes the BAN look like a normal IP network, it is natural for the BMG to terminate all BAN internal mobility signalling if indeed it is not terminated at some device further down the network. This gives a picture where BRAIN specific functionality is pushed to the BMG and BAR. Of course a larger subset of the nodes in the BAN may contain BRAIN specific functionality. It is important that this functionality supports scalability and robustness. The BMG is used here as a specific example.

In this view of the BAN, BMGs (and possibly other entities, such as the Anchor Point in the BCMP, detailed later), are 'responsible' for subsets of the address space. This essentially becomes a choice about which addresses are advertised by a BMG to external networks. This choice defines one of the most important attributes of the solution. If the pool of addresses is not localised (all BMGs advertise all addresses), then the addresses have no implied locational significance (i.e. a particular address is not assumed to belong to a particular area of the BAN). This makes scalability to large address spaces harder, as routing state for all addresses will permeate the network. Alternatively, if a block of addresses is owned by a single BMG, then there is no redundancy and therefore no inherent resilience (however, the addresses have strong locational significance). Generally the mapping will be between these extremes.

Where a BAR is reachable from multiple BMGs, mobility support may need to be extended. For example, if a MN is distant from its 'home' BAR, routing may be direct from the ingress BMG (handover updates information in multiple BMGs). Alternatively, routing may be to the home BAR and then around the edge of the network to the MN's current location (the edge-mobility model). On the other hand, if the MN has not moved from its initial position, routing from the ingress BMG may be direct (if a route is known) or to the 'home' BMG and then around to the preferred BMG for that area of the BAN.

If RSVP proxies are used (one of the QoS extensions, see section 4), then the MN or BAR needs to know which end-point to use. So, where there is a choice of multiple BMGs through which downlink traffic can enter the BAN, it may be necessary for the ingress BMG of a particular traffic flow to be known.

3.5 Paging

3.5.1 Introduction – Idle and Stand-by Modes

A MN in the *active* mode is one that is sending and/or receiving packets via its network interfaces. Two further separate but interrelated modes are defined in order to optimise its mobility support, which can be coarsely characterised as follows:

?? **stand-by mode** - its main goal is to save battery power in the MN, by allowing a MN (or subparts, e.g. its radio) to 'switch off' during a sleep period. Typically such a *stand-by MN* wakes up at well-defined times, during which the network can reach it. The stand-by mode enables link layer power management and so is only relevant to the particular wireless technology. Hence it is not considered further in this section (see section 5, annex A3.2 and [1.3]).

?? **idle mode** - its main goal is to reduce location update signalling over the air and in the BAN, by tracking the location of an *idle MN* less accurately than for an active MN. The idea is to define a *Paging Area*, consisting of several BARs that correspond to some geographical area. Only when an idle MN moves into a new Paging Area (PA) does it have to send a location update to the BAN, whereas an active MN must send a handover update message every time it moves onto a new BAR. Clearly, the network must be able to re-activate the MN (for example, if a correspondent wants to communicate with it); the process by which it does this is called *Paging*.

In this section we're mainly concerned with paging in the network, i.e. how the BARs in a paging area are alerted that a MN in that paging area needs to be woken up.

3.5.2 Existing Solutions to Paging

There are some important differences between the various existing paging proposals, for example:

?? Where the location information is stored - in [3.4] it's centrally (in the highest mobility agent of the domain), whereas in [3.5] it is distributed throughout the BAN

?? Whether the paging areas are configured by the BAN operator or individually by the mobile itself [3.6].

?? How the MN informs the BAN that it is moving into idle mode - in [3.4] an explicit signalling packet is sent, whereas in [3.5] it's simply the absence of routing refresh messages.

?? Where the paging is initiated from - e.g. the Foreign Agent [3.7] or a specialised paging agent [3.5].

Most proposals (cf. annex A3.2.2) rely on *multicast* to distribute the paging request to all the BARs in a paging area, i.e. each paging area corresponds to a multicast address. (If the BAN does not support multicast, then the same effect can be achieved with a series of unicasts.) The BAN thus stores a mapping between the identifier of the idle MN and the multicast IP address of its paging area.

3.5.3 BRAIN Paging proposal

3.5.3.1 Messages for Paging and Location Updates

We recommend that explicit messaging is used by the MN to inform the BAN that it has moved into idle mode, and also when it is about to detach from the network - rather than (for instance) the BAN interpreting the absence of uplink data as an implicit signal. This approach is an efficient and effective way of ensuring that the BAN's view of the MN's state is correct and consistent. It also allows a MN to decide for its own reasons to go idle, for instance the absence of a TCP connection.

However, to cope with a MN failing or switching off unexpectedly, the paging entry should eventually time out. This means the MN must just occasionally refresh the entry (through sending a location update); the frequency required is a design decision. The MN also needs to send an update message when it detects that it has changed its paging area. This could re-use a standard unplanned handover message, for example.

Section 5.3.2 deals further with messages over the air interface, and in particular whether they're transmitted at the link or IP layer.

3.5.3.2 Paging Initiation

Data destined for an idle MN will travel through the BAN until it reaches the point where the idle MN's location is stored and hence from where paging can be initiated. We recommend that paging is initiated from the last BAR where the MN was active, rather than (for instance) having one paging node per PA. This distributes paging information and can allow messaging (paging and location updates) to be constrained to the edge of the network, which aids scalability of the solution. It also speeds up the paging of a MN that hasn't moved. Further, it de-couples the paging procedure from path updates (i.e. it makes paging transparent to the routing protocol), because the page is initiated from a 'known point' that exists whatever the path update mechanism.

However, the BAR is now a single point of failure: if a BAR fails or is taken down for maintenance, then any MNs with their paging entry at this BAR become unreachable. In order to improve the resilience, a suggestion is that there is a 'backup' central paging controller. It needs to be reliable but doesn't need to be very scalable (since it only gets hit when a BAR fails).

The BAR (and any 'backup' central paging controller) store a mapping between the identifier of the idle MN and the multicast IP address of its paging area. Annex A3.2.3 considers further what this identifier should be (in most proposals it's an IP address), but in any case this issue should not have a strong impact on the whole paging scheme, since it could just be thought of as a parameter of the paging request.

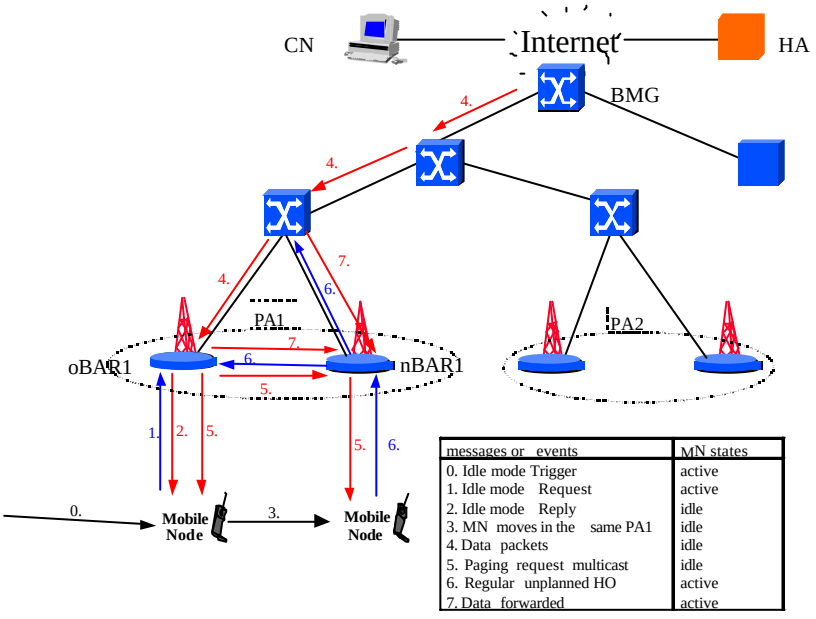


Figure 3-6: Paging is initiated from the last BAR that the MN was active on

3.5.3.3 Paging Areas

The size and shape of the paging areas needs to be optimised to minimise the overall signalling traffic; the trade-off is between location updates (reduced by large paging areas) and paging requests (reduced by small paging areas).

One issue is whether the paging areas are configured by the network operator or individually by the MNs themselves. We recommend the former; the latter would add considerable complexity, especially to achieve resilience. However, some MN-control can be provided through overlapping paging areas; for

example, this could be beneficial when a MN would otherwise frequently move between two paging areas.

The operator will determine the size and shape of the paging areas, based on a large number of factors, such as the MN's traffic and mobility patterns, the link layer's capabilities, and the network's topology. The operator will be guided by simulations and deployment experience. In practice, the important point is that overall our paging solution allows a lot of flexibility within a single consistent framework, for example allowing adjustment of the paging areas 'in the field'.

3.6 BRAIN Candidate Mobility Protocol

3.6.1 Introduction

Following our critical analysis of existing mobility management protocols, we have concluded that there is - and can be - no single scheme that is best in all scenarios. The solution which is best depends on the BAN provider's business model and deployment environment. In some scenarios, the existing solution can be improved through a protocol we have developed within the Project: the BRAIN Candidate Mobility Protocol. In particular, it fulfils our requirements for Handover (section 3.2), Paging (section 3.5) and 'Scalability and Resilience' (section 3.3), and it improves on existing 'gateway-centric' protocols for Path Updates (section 3.4). In this Section the key features of the protocol are briefly described and discussed. For more details, including message formats, please read the annex A3.5. On-going work in the MIND project will assess its suitability further.

3.6.2 Key features of BRAIN Candidate Mobility Protocol

Figure 3-7 illustrates a BRAIN network that implements the BRAIN Candidate Mobility Protocol (BCMP). The network consists of legacy IP routers with added mobility aware functionality in just two types of nodes. Anchor Points (ANP) own and allocate IP addresses, authenticate users, maintain user records, and tunnel packets towards Mobile Nodes (MNs). Brain Access Routers (BARs) terminate tunnels from ANPs and forward packets to/from mobile hosts. BRAIN Mobility Gateways in the Candidate Protocol need not have mobility specific functionality - their role is to shield the rest of the BAN from the exterior routing protocols and distribute traffic within the BAN to the correct ANPs.

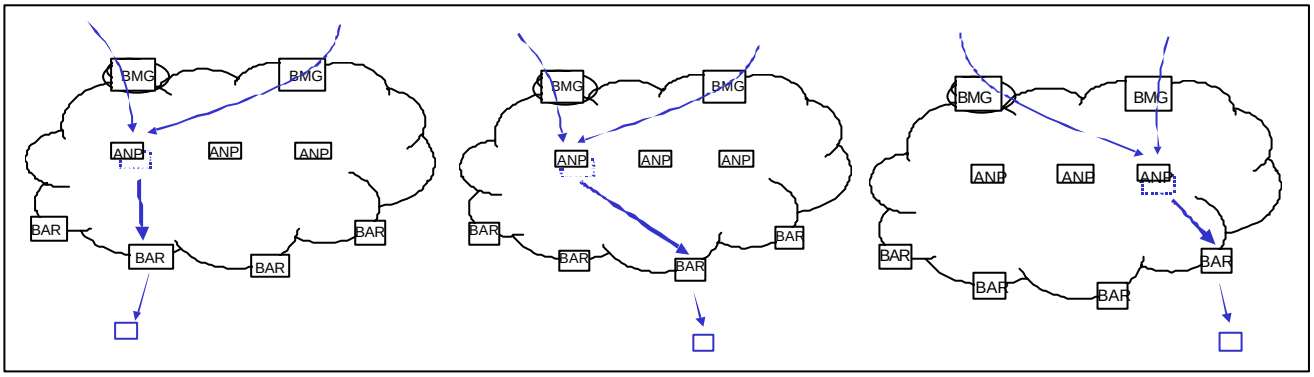


Figure 3-7: Outline of the BRAIN Candidate Mobility Protocol in a BAN

Anchor Points have globally routable address space and they allocate IP addresses to Mobile Nodes when they log in to the BAN. This address is kept constant, despite handovers. The pool of IP addresses owned by an Anchor Point is advertised using legacy IP routing inside the BAN and towards external IP networks. This ensures that packets addressed to a Mobile Node's locally obtained address are prefix-based routed to the Anchor Point that allocated the address. Anchor Points, in turn, tunnel packets to the BAR where the destination MN is located at the moment. Anchor Points must maintain up-to-date location information of MNs they have allocated an address for and must update this information when 'their' MNs change BAR (i.e., handover).

3.6.2.1 Log-in: Address Management and Security

When the MN first contacts the BAN it must execute a login procedure. First it sends a login request message to the BAR at which it has appeared. In this request the MN provides login and security information for an external AAA procedure. The BAR selects an Anchor Point for the MN according to a policy specified by the operator and forwards the login request to it. The MN need not be aware of the policy and of the internal structure of the BAN. The selected Anchor executes the AAA procedure to

identify and authenticate the MN and allocates a globally routable IP address and a new Session Identifier for MN, which is a temporary identifier for the MN. The session id is discussed further in annex A3.5.1.

The session id, key and IP address are sent back to the MN in a login response message.

3.6.2.2 Handover and Path Updates

The BCMP includes an optional handover preparation phase to ensure fast and smooth handover. If the MN knows in advance to which BAR it is moving to (i.e., planned handover), then it can build a temporary tunnel from the old to new BAR, for example. For other likely actions see section 3.2.

The handover execution procedure is the same whether or not there's been a preparation phase. The main task is to inform the Anchor Point (ANP) about the handover and remove the temporary tunnel. It is initiated by the new BAR by sending the handover message to the ANP. In response the ANP will redirect the tunnel to the new BAR and notify the old BAR. Upon receiving this notification the old BAR can safely remove the temporary tunnel because it will certainly not receive more packets from the ANP. This concludes the handover.

3.6.2.3 Inter-Anchor Handover

If a MN moves far away from its Anchor Point then the tunnel between the ANP and BAR may become very long. In order to avoid long tunnels, the protocol allows (but does not mandate) the network operator to request that a MN changes ANP. This improves routing efficiency in the BRAIN network, in exchange for exposing mobility toward the Internet: the change of ANP requires changing the MN's IP address, that is a global mobility event. Alternatively, operators may choose to accept long tunnels between ANPs and BARs in order to completely hide mobility from external networks.

3.6.2.4 Paging

Paging support in the Protocol allows MNs to enter idle mode and to reduce location update signalling load inside the BAN. Downlink packets are tunnelled to a MN's last known BAR, which knows that the MN is idle and so initiates the paging process.

3.6.3 Discussion of BRAIN Candidate Mobility Protocol

Here we briefly discuss how the BRAIN Candidate Mobility Protocol (BCMP) compares to the requirements identified earlier in this Chapter.

?? **Handover management framework:** BCMP is fully compliant with the handover framework (section 3.2). The signalling which relates to selection amongst multiple CARs, and which precedes the handover execution, is an optional feature in BCMP.

?? **Path Updates:** In general, BCMP could be characterised in the following way. The protocol is 'gateway-centric' (see Figure 3-4). It decouples fixed and mobile routing, which respectively use standard prefix-based routing to the Anchor Points and MN-specific tunnelling from the ANPs to BARs. It uses soft-state information, i.e. there are periodic refresh messages. For a 'normal' handover, a Path Update originates from the MN and then (only) travel between the BARs and ANP, but an inter-anchor handover is a global mobility event. BCMP requires a limited number of 'BAN specific' nodes, i.e. the ANPs; it is a design decision where (and how many) ANPs are sited in the BAN, and this is currently being explored through simulations.

?? **Resilience and scalability – multiple BMGs:** BCMP decouples some of the functionality in the BMG, to leave the BMG as an Internet peering point and introduces the ANP to manage the mobility specific functionality. Essentially, the earlier discussion (section 3.3) applies equally to the ANPs as well as to the BMGs. As standard Internet peering points, the BMGs fulfil all the scalability and resilience requirements. Some functionality, such as location tracking, resides solely in the Anchor and issues such as knowledge of the ingress BMG (as opposed to Anchor) are for future study. Multiple ANPs provide a controlled way of offering scalability in both throughput and geographic area. Although the BCMP does offer some resilience, it does not do so transparently because the terminal must re-login to an alternate ANP or another AP must be able to take over the address space of the failed AP without impacting the terminal.

?? **Paging:** The paging mechanism fits in with the earlier recommendations (section 3.5). Paging is initiated from the last BAR that the MN was active on. A MN sends an 'idle mode request' message to the BAR its currently attached to, and goes into idle mode when it receives a Reply. The same message is used as a refresh. When an idle MN detects that it has moved into a new paging area, it sends a regular unplanned handover message (HOFF). An idle MN sends a regular LOGOUT message just before it switches off.

We have simulated the BCMP and shown its correctness of operation. On-going work is exploring issues such as where to position the Anchors in the BAN, and the impact of inter-anchor handovers.

Figure 3-8 shows a snapshot of the simulation, showing the BCMP performing a planned handover. More simulation results can be found in annex A7. (The mobile has performed a planned handover from the old AR(12) to the new AR(13) and is notifying (white packets) the Anchor (which is unchanged) of the new AR to which its packet must be forwarded and is also notifying the old AR to stop sending packets over the wireless link since these packets are lost)

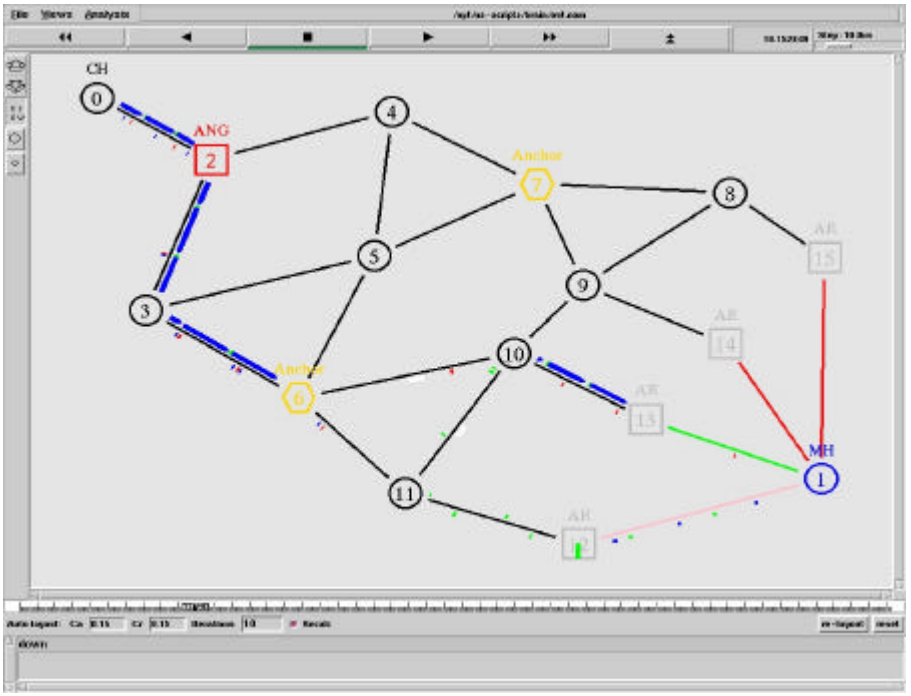


Figure 3-8: Snapshot of Simulation of BCMP- Planned HO

4 Quality of Service

4.1 Introduction

The Internet supports a large number of users with a simple, best effort network, which ensures that all users receive a share of available bandwidth. This simple, flexible network has supported the development of a wide range of data applications. However, Quality of Service (QoS) within the Internet is needed if it is to successfully carry a significant quantity of real time services such as audio or video-conferencing. A network that provides quality of service transmits some packets preferentially to achieve, for example, low transfer delay. Parameters that can be controlled include packet delay, packet loss, packet errors, available bandwidth and inter-packet delay (jitter). These parameters may be controlled absolutely ("guaranteed") or they may be controlled within some probabilistic framework.

Solutions exist that address particular aspects of the QoS problem. The Session Initiation Protocol enables users to be contacted and agree on suitable coding schemes for multiple flows within a session. Once the session is established, the Real Time Protocol ensures that packets are replayed to the user at the correct time, thus overcoming jitter and stream synchronisation problems. The Real Time Control Protocol provides feedback about the quality experienced by a session. If problems are detected, RTCP can be used to trigger SIP session re-negotiation. Stream synchronisation can even be achieved between different sources if they have time synchronisation through the Network Time Protocol. These (layer 5) protocols are well established within the Internet. However, these alone do not solve the real-time service problems as they cannot guarantee that packets in any one flow are delivered across the Internet in a timely fashion. The Internet requires an end-to-end (layer 4) mechanism to ensure timely packet delivery through the network. This can only be achieved if there is timely packet delivery at every router within the Internet (layer 3). It is these mechanisms that are the focus of this study. However, there is potential for interactions between the transport layer QoS and the higher layer QoS mechanisms. It is the purpose of the well defined Enhanced Service interface and interface implementation (5.2) to provide these higher layer functionalities and manage the interaction issues. These are indicated in Annex A4.2.1.

Many lower layer mechanisms also exist to provide QoS. In general these link layer specific mechanisms cannot achieve end-to-end quality of service for the user. These mechanisms may however be used to provide QoS to the network layer. This is of particular importance over any wireless interface. To achieve network layer QoS across a wireless network requires a link layer support. Thus the mobile nodes and the BAR routers need to be able to map network layer QoS to link layer QoS. A specific example of how this could be achieved, the IP2W convergence layer, has been studied within this project and is reported in section 5. There is a potential for interaction between these two mechanisms, which is discussed further in section 5.3.3, and Annex A4.2.2.

Whilst transport and network layer QoS has been studied within the Internet for many years, there has only been limited deployment of any network QoS mechanisms. However, recent developments within the IETF have produced a framework, the ISSLL (Integrated Services over Specific Link Layers) QoS architecture, which in our opinion addresses many of the problems (such as scalability) of previous solutions. In particular the solution is able to use a Differentiated Services (DiffServ) network layer mechanism. This provides scalability and reduces the complexities that were the original problems associated with the Integrated Services (IntServ) architecture. ISSLL can be seen as expanding the DiffServ architecture, enabling it to be used to provide end-to-end service to the user. Thus the solution provides flexibility of service provision and the possibility of firm end-to-end QoS guarantees. It clearly separates the network (layer 3) QoS providing mechanism from the (layer 4) end-to-end signalling, thus leaving much freedom for different network providers to choose the most suitable QoS mechanisms and call admission scheme. We chose to use this framework after analysis of the basic design questions, summarised below, that must be considered by any QoS mechanism. An additional benefit of this choice is that operating systems including Microsoft's Windows, and small-scale corporate networks are now appearing with the capabilities to utilise the ISSLL solution.

However, there are weaknesses with this solution when it is evaluated against requirements if both wireless access and mobile users are assumed. The hard requirements on the QoS solution are discussed in annex A4.3.4.1. This is not surprising, as wireless and mobility issues have received no attention within the IETF QoS community to date. Wireless issues revolve around the restricted bandwidths and increased error rates of wireless links, and how techniques to overcome these problems may interact with higher layer QoS mechanisms. These are further discussed in annex A4.2.3. Mobility issues, discussed further in annex A4.2.4, relate to the behaviour of the system during and after handover. To solve the particular problems identified, a selection of possible extensions has been studied. Some of these extensions have

been specifically developed under this project, and these in particular are highlighted. Many of these extensions can be provided transparently within the network, however others have larger implications. Many can be deployed as independent improvements to the network, others interact in some fashion. The most suitable choice of extensions depends in part upon the business environment of the network operator. These issues are discussed briefly in the final section, where we show clearly how the various extensions can be used to solve the weaknesses of the baseline ISSLL architecture.

4.2 Base-Line Architecture

The base-line architecture chosen for the BRAIN network is the ISSLL architecture, where the wireless part of the network is considered as the ISSLL access network, implying a per-flow reservation at that hop, and the fixed part of the network is considered an ISSLL DiffServ core network. This is described below. This architecture has been chosen as the result of analysis of the key issues that affect the design choices. Whilst extensions can be used to enhance this basic architecture, any BRAIN network must be able to provide the services as expected by this architecture to any terminal that utilises the defined interfaces.

4.2.1 Basic Design Choices

There are a number of different ways of providing the generic functionalities needed within any QoS architecture. Before detailing how we provide these within BRAIN, we first highlight some of the rational behind key choices that have been made. Fuller details of all design issues considered may be found in Annex A4.3.1

QoS is only useful, and therefore most likely to be paid for, if it exists on an end-to-end basis. This does not mean that the QoS mechanisms used to provide a particular guarantee need to be the same across the network. Thus, here an end-to-end solution is chosen, and in particular the end-to-end signaling protocols used are standard Internet protocols. This approach contrasts with that of a number of large network providers who charge for guarantees associated with, for example, virtual private networks, without concern for the end user. This unfortunately has led to quality of service mechanisms being developed in ways that may be inconsistent with end-to-end design which could therefore restrict the future development of innovative services on the Internet.

Many wireless networks, such as GSM style networks, provide close coupling between wireless network and the application. This is done to achieve maximum efficiency and optimize performance. For example, this coupling allows the link layer to provide different loss management techniques for voice (which prefers random bit errors) and video (which prefers whole packet losses). This approach is not taken here, as it breaks the strict layering principles. No attempt is made to pass wireless specific information through the IP stack. This approach does not prevent link layer manufacturers from making improvements to the basic performance of the link layer. For example, they could improve the loss rate across the link through the use of schemes such as Forward Error Correction (FEC) or, where it does not break any requests for fast packet delivery, Automatic Repeat Request (ARQ) could be used. Similarly, IETF standards show how the link layer can perform RTP/UDP/IP header compression to save on bandwidth requirements. This does not require that the link layer know that the data it is attempting to compress is RTP data. These improvements are transparent to the network layer, but will be seen as a benefit by the users. It has not been possible to evaluate the loss in efficiency that results from adherence to strict layering principles, but it is believed that it is a small concern in broadband wireless networks.

QoS may be achieved through per flow reservation. Here an application queries the network to discover if the QoS requirements can be achieved. Reservations make best use of resources allowing better planning of the network usage, and giving a more reliable QoS. However, there is a large overhead associated with this, as signalling messages are required and there is a delay before applications can start to send data. In the alternative prioritisation model, clients mark their packets to indicate a “premium” service requirement. The service may be used at any time, but the performance provided will be less predictable and may suffer from network congestion. Prioritisation is typically used with Service Level Agreements (SLAs) that may be general, or defined on a per-user basis. The solution chosen provides both reservation based and prioritisation services.

Per-flow traffic management means that the application's traffic is granted resources and protected from the effects of traffic from other users in the network. This enhances the quality of the service experienced by the application, but also imposes a burden on the network which needs to maintain state for each flow and to apply independent processing for each one. In the core of large networks, where it may become necessary to support millions of flows simultaneously, per-flow traffic handling is not practical. Alternatively, when traffic is handled in aggregates the state maintenance and processing burden on

devices in the core of a large network is reduced significantly. However, the quality of service is no longer independent of the effects of traffic from other sources. Over-provisioning of resources to the aggregate QoS traffic can offset this effect. However, this approach tends to reduce the efficiency with which network resources are used. Whilst we do not require any specific network internal QoS mechanism, the DiffServ aggregate mechanism is considered highly suitable because of its scalability and potential simplicity, and it is assumed throughout this text.

QoS treats some flows preferentially to others, and this implies the ability to reject flows. Admission Control functionality, used for both prioritisation and reservation based services, may exist at various places within the network. In the basic architecture, edge-routers (the BAR and BMG nodes) process resource requests. These nodes use their local knowledge of their current state to make a decision on behalf of the core of the network. This is the simplest solution to admission control and is inherently the most scalable. However, it is not the most suitable for all network topologies, as no global knowledge about current network state is used in the admission control process. This can lead to inefficient use of network resources, or poor QoS guarantees. Alternatives are a centralised approach and a hop-by-hop approach. The relative merits of these approaches are discussed in the Annexes A4.3.1.8 and A4.3.1.9, and they are also introduced as extensions to the basic architecture in Section 4.3. These different schemes, both inherently less scalable than the edge solution, can be used without any changes to the interfaces between mobile node and network.

4.2.2 Description of Base-Line Architecture

The BRAIN QoS architecture, as indicated in the figure below, is based on the ISSLL framework. The Resource Reservation Protocol (RSVP) is used with the Integrated Services message set as the signalling protocol for explicit resource reservations. These RSVP messages are exchanged end-to-end across the network. Within the BAN, RSVP messages are only interpreted at the network edges – the BAR and BMG. These nodes map RSVP reservation requests to DiffServ forwarding classes, keeping a record of the session identifier and the required DiffServ class. Once data begins to flow using the reservation, these edge nodes (BAR and BMG) will map the packet headers to session identifiers in the packets and confirm that the correct DiffServ Code Point (DSCP) is being used. Once within the network all packets are forwarded according to standard DiffServ operation.

In addition to the per-flow signalled reservations, the architecture also allows normal DSCP marking to support prioritisation QoS for applications that are not able to quantify their resource requirements. An important decision in the architecture is to leave the outbound flow marking to the mobile node. This is important as it allows the MN to use IPSec payload encryption. The inbound flow may be marked at source, but more likely marking occurs at the gateway nodes. The gateway determines the correct marking for the packets as Service Level Agreements (SLA) provide a mapping between IP packet header information and the relevant DSCP. This SLA information may be retrieved using the COPS protocol from a network policy server.

For a full description please see annex A4.3.2.

4.2.2.1 Network Nodes

The BAR is the first (last) IP-based node within the BAN to which a flow originated from (terminating to) a mobile node arrives. The BAR is in charge of resource co-ordination for the access points[1] linked to it.

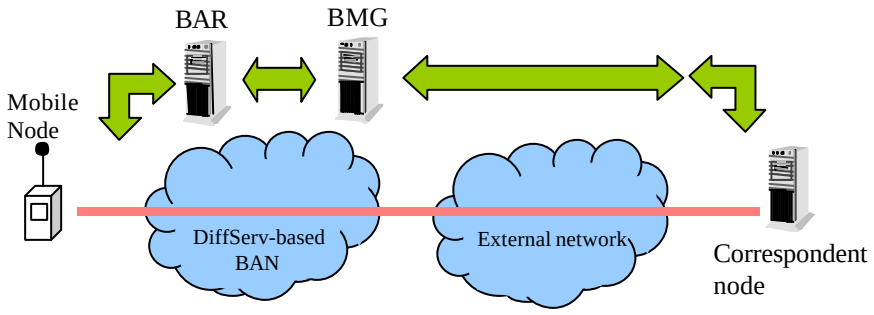


Figure 4-1: QoS in the Network Nodes

[1] By Access Point we denote a layer 2 only device through which IP packets are forwarded transparently.

The BAR and BMG are in essence DiffServ edge nodes – the functionality, such as traffic policing, of such nodes is described in annex A4.3.2.3. They must support RSVP signalling and map the signalled IntServ/RSVP requests to a suitable DiffServ Per-Domain Behaviour. The BMG has the additional responsibility of allocating the appropriate DS behaviours to inbound prioritisation traffic based on service level agreements. The BMG should also do strict shaping of incoming flows.

Once data is within the BAN, internal routers forward packets according to normal IP forwarding mechanisms and the DiffServ processing. The need for this DiffServ processing can be avoided if the BAR and BMG perform proper shaping of flows admitted into the network. In this situation, the BAN can be over-provisioned and built with ordinary routers. Conversely, in a larger BRAIN network, the internal routers may all be DiffServ capable, and some may even be RSVP-enabled to more closely control the resource sharing among flows.

4.2.3 Error Reporting

A key requirement for applications is that they receive adequate error reporting, to facilitate application adaptivity. The errors under consideration include those created by a sender breaking its contract, and those that occur as a result of network faults including congestion and temporary loss of QoS during the handover process.

RSVP based Reservation QoS has error reporting mechanisms available within the protocol. However, these messages, which indicate a network fault, can only be generated at RSVP aware nodes, and are only generated when the reservation refresh process fails. This may mean that error conditions exist for some time before the problem is identified and reported. When RSVP is used, as suggested below, in hard state mode with local path repair to handle mobility events then errors would not be reported as a result of the handover process.

DSCP re-marking can be used to provide some indication to a receiver of problems with a transmission (for example that the data broke the contract, or that the requested service is not currently available). The amount of information that can be conveyed in this way is limited.

A more general mechanism is required to provide feedback for any DS marked (reservation or prioritisation) traffic, regardless of where in the network problems occur. Higher layer protocols are generally assumed to monitor received QoS, and to provide feedback to the data to provide adaptation – this is a standard part of TCP, and is the purpose of the RTCP element of the RTP. However, this leads to a long time delay between QoS disruption occurring and the data source being able to adapt to the situation.

This time delay can be reduced, if network layer mechanisms, such as Explicit Congestion Notification (ECN), are used with Random Early Detection (RED). This mechanism provides advance notice to a receiver that congestion is expected to occur, giving the receiver time to notify the data source before network congestion actually occurs. ECN plus RED is becoming a central element in reporting of congestion in a network. ECN interaction with TCP has been studied, and work is ongoing towards using ECN with UDP flows.

Whilst the use of ECN is recommended within the basic architecture, two assumptions are implicit in this approach. The first is that all errors are a result of congestion. This is not true in the wireless environment. Better performance can be achieved if a distinction is made between congestion and loss. Within the BRAIN project, we have assumed that the link layer may provide a local (fast) mechanism, which may be used by the mobile terminal for this purpose. This aspect is not considered further within this discussion. The second assumption of ECN and RED is that congestion grows slowly within a network, so that it is possible to notify the data source before congestion becomes serious. Within this project we have identified that this is not true in a network with mobility support, as mobility may suddenly lead to congestion appearing at a router. Within the project we have considered mechanisms to overcome this problem, using the Internet Control Message Protocol to send error reports directly to the data sender. As of date, we are unable to recommend such an approach as outstanding problems remain with a possible explosion in the number of control messages being generated at exactly the same time as the network is experiencing congestion.

Finally, although a range of different error reporting mechanisms is included, this complexity is hidden from the application developer through the ESI (section 5.2) and the associated layer implementation.

4.2.4 Evaluation

For a discussion detailing how the transport/network layer solution works with the other quality protocols to provide an overall solution to support real-time interactive voice, the reader is referred to the worked example section.

There are a number of assumptions and key requirements on which the BRAIN QoS mechanism is based. Fuller details can be found in annex A4.3.4.1. In addition to the hard requirements upon a system, there is also a selection of evaluation criteria against which any potential solution should be evaluated. These evaluation criteria are critical in choosing a QoS solution. Fuller details of these can be found in annex A4.3.4.2. These requirements and evaluation criteria have driven the development of the base-line architecture.

Analysis of the base-line architecture against these requirements shows that its strengths are that it:

1. Allows both prioritisation and reservation based QoS.
2. Uses standard protocols to support generic IP hosts.
3. Is scalable because of DiffServ flow aggregation.
4. Can provide hard guarantees for services such as voice, with low overheads.
5. Can be simply implemented in routers using class based queuing.
6. May be provided with only QoS aware BAR and BMG nodes.
7. Provides QoS renegotiations.
8. Is in line with IETF architectural principles.
9. Hides network internal mechanisms, making this an easily evolvable solution.
10. Provides limited support for handover.
11. Is resilient against network failures.

However, as a result of wireless and mobility issues, specific weaknesses of the proposed solution are:

1. The strengths of service guarantees that can be achieved by the base-line architecture are limited because of the call admission architecture

The edge nodes can only make approximate admission control decisions. In an access network with either little capacity for statistical multiplexing effects or restricted bandwidth, the quality of these decisions will be weak. The possibility of mobility further weakens the strength of such decisions. The standard IntServ classes are not well suited to the wireless environment.

2. The base-line architecture does not minimise the quantity of signalling required

Standard ISSLL RSVP operates in a soft state mode, with refresh messages every 30s. These messages are large. Mobility may even trigger the mobile node to send more RSVP messages to repair a broken path.

3. The base-line architecture does not support seamless handover

Quality reservations will be disrupted in the base-line architecture, until the refresh mechanism can restore the reservation. This can result in long disruptions to the quality experienced by a data flow. The disruption can be seen to have several different aspects. Firstly, link layer reservations may be easily created as part of the handover process to ensure that QoS is quickly established on the wireless link, which is typically the weakest link. In this situation however, the BAR needs to be able to map incoming layer 3 packets to the appropriate layer 2 reservation. This does not occur until the network path QoS is repaired. Ideally, this PATH repair should happen quickly. Finally, during the actual handover process packets may be tunnelled between old and new BARs. For the user to experience no QoS disruption, these packets must also be routed according to the original QoS reservation.

4. It does not provide any reservation based network layer QoS in the absence of transport layer end-to-end QoS signalling

If the end-to-end signalling is not available, for example if the CN (or its legacy voice application) does not support RSVP, or the RSVP messages are badly routed through a QoS unaware (possibly over-provisioned) network, then reservations cannot be established. This requirement is of particular interest if the access network is bandwidth limited, or if QoS needs to be provided within a wireless network without significant progress in the deployment of IP QoS.

A large number of extensions have been proposed to solve these specific limitations. These extensions provide different levels of functionality. However, provision of this functionality comes at a cost, which may be increased network complexity, or it may be that it requires changes to either mobile terminals or even the end-to-end network protocols. Thus, the choice of suitable extensions is a choice for the network

operator and their business models. However, because all stem from the same base-line architecture, interoperability is assured between terminals on different BRAIN networks. These extensions are discussed in the next sections.

4.3 Solutions to weaknesses in the base-line architecture

This section provides a brief description of possible extensions to the base-line architecture. The following section then shows how these extensions can be used to address the shortcomings of the base-line architecture, and in particular highlights any potential interactions between different extensions. Particular detail here is given to extensions that have been developed by the BRAIN team. Whilst many possible extensions have been reported in the literature, we have focused on the sub-set that we believe is most important.

4.3.1 QoS Context Transfer

Problems solved:	seamless handover, also facilitates other extensions.
Scope of impact:	changes required at mobility aware routers that maintain QoS state.
Status:	Being considered by the IETF SEAMOBY group, to which BRAIN members contribute

A context transfer protocol transfers state information about the mobile's QoS requirements during handover from old to new BAR. This exchange is triggered by hand-over indications received from the link layer. An example, developed by the BRAIN team, of how the context transfer protocol can be loosely coupled to link layer indications is provided in annex A4.4.1.1. QoS context information should also be exchanged between tunnel nodes, such as the anchor points in the BCMP. The relationship between the BCMP and QoS is discussed further in annex A4.5.3.

The protocol needed for this procedure, and the parameters that must be exchanged, are for further study. The context transfer protocol will require support in all mobility-aware nodes within the BAN. This is part of the work of the IETF Seamoby working group, to which BRAIN is actively contributing. Within Seamoby, context transfers are discussed in wider terms including security information and header compression as well as QoS related information. This protocol is a pre-requisite for certain extensions including Modified RSVP local path repair.

This protocol facilitates seamless handover. It is assumed that the layer 2 wireless QoS reservations will be restored as part of the handover process. The context transfer protocol will provide the new BAR with sufficient information for it to map the incoming IP packets to the different wireless reservations. This will quickly restore QoS over the wireless link, which is usually the weakest link in the communications path.

Additionally, when RSVP local path repair is used, as discussed as part of extension 4.3.3, hop-by-hop call admission, then the Context Transfer Protocol enables a reduction the signalling load over the wireless network.

4.3.2 Bandwidth Broker

Problem Addressed:	Improves strength of QoS guarantees
Scope of Impact:	Changes required at admission control nodes, and this introduces new nodes
Status:	Requires further study and progression within IETF

Bandwidth brokers make the admission control decisions[2], on behalf of BAR/BMG nodes. A bandwidth broker has global state knowledge, because it receives each admission control request, and so can make a strong admission control decision. As part of this decision, it can also take likely mobility patterns into account. Bandwidth brokers may be a centralized unit. A similar effect may be gained if the bandwidth broker functionality is distributed between adjacent edge routers. This approach would give some global benefits, in particular enabling possible mobility patterns to be taken into account, whilst removing the "single point of failure" that a centralized node represents. Additionally, such a locally distributed broker could be more scalable and reduce the network-signalling load. Current definitions of bandwidth brokers within the IETF do not provide support for DS awareness, however the Internet2 Qbone design team have made significant progress in bandwidth broker implementation. This is a simple, but highly worthwhile

[2] Note that there is no admission control process for best effort traffic

extension requiring support in the nodes that perform admission control. This is discussed in more detail in annex A4.4.2.

4.3.3 Hop-by-hop Call Admission

Problem Solved: Improving strength of QoS guarantees

Scope of Impact: All routers within the BAN

Status: Standard part of ISSLL framework

Each router is responsible for its own admission control decision. This can lead to stronger service guarantees, especially at the edges of a network. This is a standard feature of the ISSLL framework, and is discussed in more detail in annex A4.3.1.8. However, additional extensions can be used with this to improve the performance when mobility is considered. Such extensions provide a fast repair of the reservation when the data path through the network changes as a result of mobility.

4.3.3.1 Protocol Coupling

Problem Solved: seamless handover - fast restoration of network reservations

Scope of Impact: Changes required at mobility-aware network nodes

Status: Loose coupling is assumed in RSVP. Tight coupling needs further study

Reservation-based QoS implicitly assumes a fairly stable path across the network. In the dynamic mobile environment performance is less than optimal. Changes to routes are only reflected in the reservation after refresh messages have passed along the new path, which can have a high latency end-to-end from mobile to correspondent node. By enhancing the QoS mechanism for the mobile environment, local path repair is possible and changes to the reservation are localised to the area affected by the change in topology.

The micro-mobility and QoS signalling mechanism are coupled either loosely via a triggering mechanism, or more tightly so the QoS and mobility information is carried by the same protocol. In the loosely coupled approach, the change in location of the mobile, and hence the updates to the routing information within the network, triggers the generation of RSVP PATH repair messages. In the tightly coupled approach, the routing and QoS information is propagated into the network at the same time, ensuring that the reservation is re-installed in the network as soon as the valid routing information is available. For the motivation behind this suggestion please refer to annex A4.4.3.

This extension is either an enhancement to the implementation of the network nodes in the loosely coupled case, or a more complex addition to the micro-mobility protocol for the tightly coupled case. The scope of this extension is restricted to the RSVP-aware routers within the BAN and maybe only those that are mobility-aware. Further details of the suggested coupling approach for the BCMP, and associated QoS issues, please refer to annex A4.5.3.

We are evaluating through ongoing simulations the effect of the degree of coupling between micro mobility and QoS signalling. Details are in annex A7.4.3 and sample results are shown in Figure 4-2.

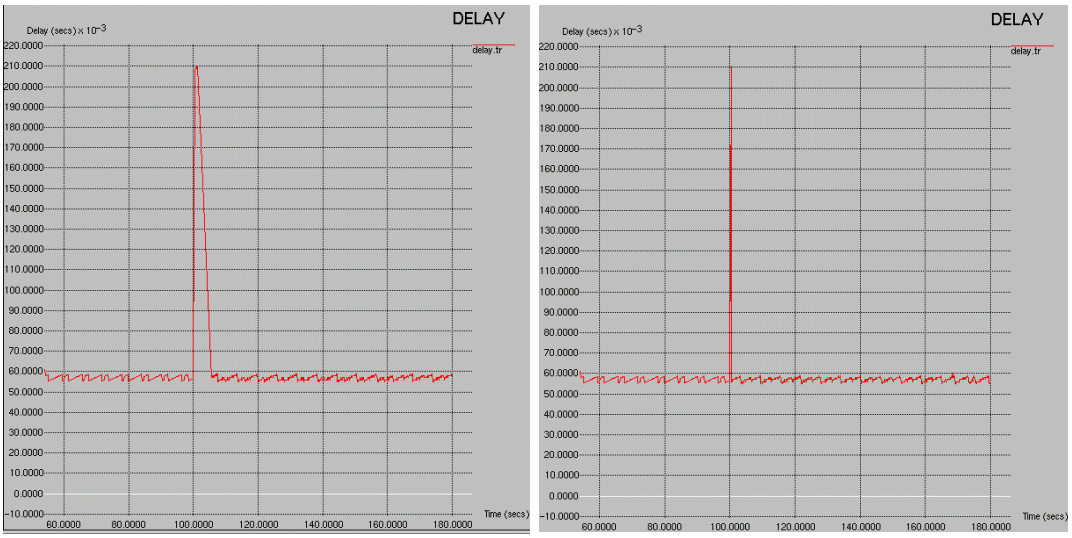


Figure 4-2: Simulation Results - Delay of VoIP Packets During a HO when De-coupled (left) and Loosely Coupled (right)

4.3.3.2 Repairing RSVP Local Path Repair

Problem Solved: Seamless handover - fast restoration of network reservations with minimal mobile signalling

Scope of Impact: changes required at RSVP aware nodes, including the MN

Status: Needs modifications to IETF draft to ensure network safety

As discussed above, the coupling between mobility and QoS can initiate the standard RSVP local path repair process. This mechanism repairs only the part of a QoS reservation that is broken, which means that the reservation can be installed faster because end-to-end signalling is not required. The signalling must not be generated until there is path stability within the network. In the standard RSVP local path repair, the mobile node would be involved in re-establishing the QoS. This extension, as discussed in annex A4.4.4, removes this overhead from the mobile by using the information transferred by the context transfer protocol mentioned above. It would require any RSVP local path repair functionality to be turned off in the mobile.

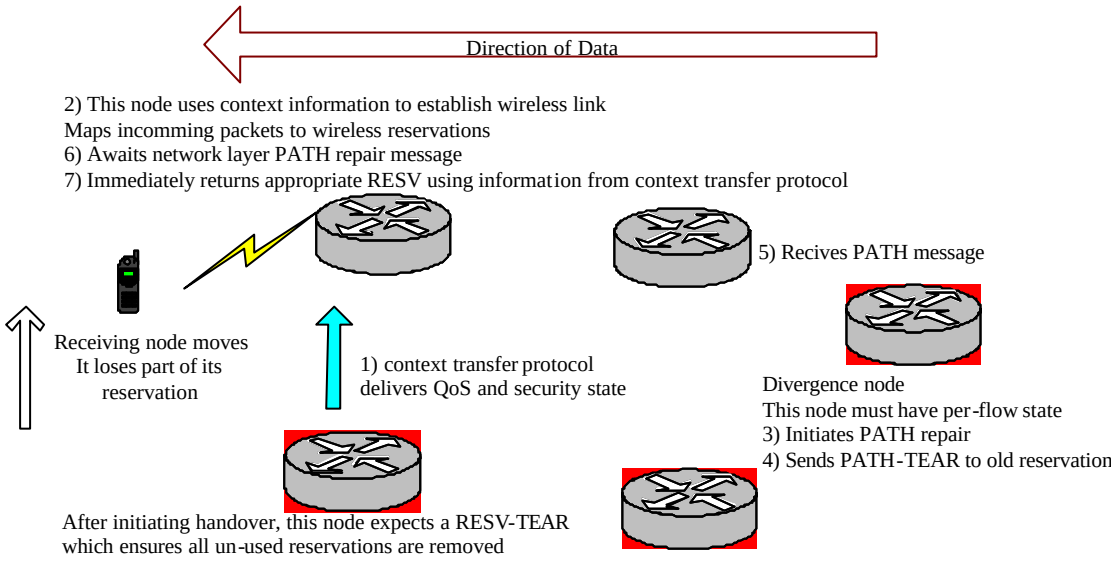


Figure 4-3: RSVP Path Repair

4.3.4 Bounded Delay DS

Problem Solved: Improving strength of QoS guarantees

Scope of Impact: All routers within the BAN

Status: Known in literature, BRAIN modifications need publication

Existing DSCPs do not provide a way to quantify maximum router delay, so cannot support guaranteed real-time services. The bounded delay class is a new DiffServ class (per-hop behaviour) defined to ensure QoS guarantees for reservation-based traffic. Whilst this service has been documented within the open literature, within this project a security weakness was identified in the original proposals. The BD service, and mechanisms to avoid any security weaknesses are considered in annex A4.4.5. All routers within the BAN must support the BD service parameters. This service would operate in addition to any standard DS services such as the expedited forwarding service for prioritisation based fast packet delivery.

4.3.5 Simplified service definitions and signalling protocols

Problem Addressed: Improving the strength of QoS guarantees, Minimising signalling

Scope of Impact: All RSVP nodes in Internet

Status: Discussion in SIGLITE, BRAIN members contributing to this

IntServ parameters are significantly larger than required in many cases. Many proposals concerning this issue revolve around the premise that only two pieces of information need to be exchanged between terminal and network - the peak bandwidth of the traffic and the service required. This is discussed further in annex A4.4.6.

This enhancement requires a change to the end-to-end signalling protocols used to signal QoS. Thus, standardisation by the IETF is needed. BRAIN partners are involved in the SIGLITE team, which may become an IETF working group addressing such issues. The suggested modifications would however significantly reduce signalling overhead, simplify admission control, simplify billing. From an end user perspective, it may even be possible to provide a more comprehensible service model .

4.3.6 Mobility Enhanced QoS Parameters

Problem Addressed: Seamless Handover, Improving strength of QoS guarantees

Scope of Impact: All nodes between anchor point and BAR

Status: Requires further study

Fixed network QoS parameters do not fully characterise desirable QoS behaviour in mobile and wireless environments. This can lead to, for example, unsuitable error correction techniques being used across the wireless interfaces, or unnecessary levels of QoS error reporting. The parameter set could be extended to include mobility-related parameters, or wireless specific information. Unless used within 4.3.10, the local BAN signalling protocol, this may require a change to end-to-end signalling protocols. Addition of such access network specific parameters breaks the layered architecture. The generic parameter set is discussed in Annex A4.4.7.

4.3.7 Hard State RSVP

Problem Solved: Minimising signalling

Scope of impact: Mobile Node and admission control entities

Status: Ability to set very long refresh times on a hop-by-hop basis is part of RSVP standard

The soft-state nature of RSVP requires periodic refresh signalling messages to propagate across the network. Where bandwidth is limited this characteristic is undesirable. RSVP can operate in a near hard-state mode by setting the soft-state timer in the RSVP module to a very high value. Whilst this mechanism is part of the RSVP standard, this project has identified that additional mechanisms are required to ensure the safety of the network and the recovery after network node failure. One such mechanism is to instruct the routers to use the presence of data traffic as implicit RSVP refresh messages, and would require support within some RSVP-aware entities within the BAN (annex A4.4.8). As the soft state refresh timer can be set on a hop-by-hop basis, it would be possible to operate RSVP normally within the network and only use hard state over the wireless interface.

4.3.8 QoS Reservations in Temporary Tunnels

Problem Addressed: Seamless handover

Scope of Impact: BARs and possibly the intermediate BAN routers

Status: Tunnelling, RSVP tunnels and DS re'marking all standardised

Temporary tunnels are typically created between the old BAR and new BAR during hand-over to prevent packet loss. Current handover schemes do not provide a means to maintain the QoS of the traffic being forwarded to the mobile. Where planned handovers occurs, reservations can be signalled between BARs. There is a trade off between the signalling overhead required to establish a temporary tunnel with reservation for each transport reservation, and the quality of service received by the forwarded traffic when it is aggregated. This is discussed further in Annex A4.5.4.

4.3.9 DS Handover markings

Problem Solved: Seamless handover

Scope of impact: RSVP nodes

Status: Ability to define network internal code point is part of DS standard, use of static guard bands for handover traffic standard is in literature

This provides a mechanism for reservation–based handover traffic to access guard bands of bandwidth, reserved purely for high priority handover traffic. Reservation marked traffic that is to be tunnelled to a new node (for example between old and new BAR routers) can be re-marked to a network internal DSCP that ensures that it experiences as high a priority as possible without disrupting any other traffic. This provides improved QoS without requiring that short-lived reservations (which produce processing and signalling overhead) need to be established.

This mechanism can also be used where there are hop-by-hop reservations within the network which are disrupted as a result of mobility. This allows the traffic to have a high priority whilst the network waits for the data path to stabilise before attemp ting to repair the network layer QoS. This is discussed in Annex A4.4.5.2.3, and is illustrated in the figure below which shows how a node must be able to identify a reservation based packet without any actual reservation, and that this node must be responsible for re-marking the packets.

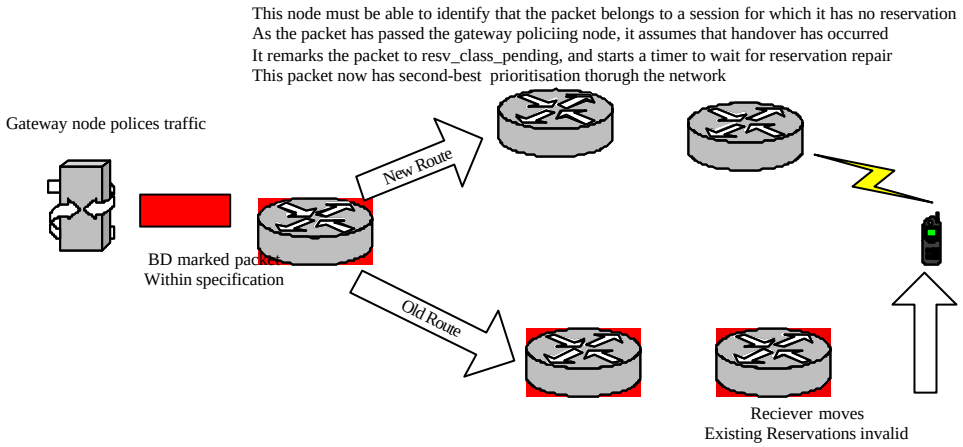


Figure 4-4: Handover Markings

4.3.10 Local BAN signalling Protocol

Problem solved:	Reservations across the BRAIN network in absence of end-to-end QoS
	Error reporting
	Seamless handover
	Minimise quantity of signalling
Scope of impact:	Mobile Node and routers
Status:	Requires further study

If the BAN has limited resources and no QoS functionality is assumed in the external networks and correspondent nodes, it is desirable to allow the mobile to provision QoS for both uplink and downlink traffic flows that will be traversing the BAN. The Local BAN signalling protocol needed for this can be included as an enhancement of the mobility protocol, for example the tightly coupled protocol mentioned in annex A4.4.9, or a specialised RSVP implementation used within the BAN. Observe that this does not preclude an end-to-end resource signalling as overlay functionality to the BAN when and if the correspondent node supports explicit end-to-end signalling. This is a complex extension requiring support in the mobile, the nodes within the BAN, and potentially within the micro-mobility protocol as well. However, if correctly designed, it has the potential to facilitate seamless handover, error reporting and signal minimisation.

4.3.11 RSVP Proxies

Problem Solved:	Reservations across the BRAIN network in absence of end-to-end QoS
Scope of Impact:	BRAIN gateways and mobile terminals
Status:	Under study in IETF, BRAIN to contribute as the proposal discussed in detail in annex solves many weaknesses of current drafts

RSVP proxies can be deployed in the BAN to proxy for end-to-end RSVP messages. For uplink reservations, the MN initiates the RSVP signalling, which is intercepted by the proxy. For downlink reservations the proxy must initiate the signalling, therefore some means to inform the proxy of the required QoS and to trigger the signalling is required. A mechanism to distinguish between RSVP sessions that need to a proxy and RSVP sessions that are end-to-end is required, and we have suggested that a flag, called an RSVP Proxy Flag (RPF) is included in the RSVP common header for this purpose. We have also identified a mechanism to enable the MN to provide sufficient information to the proxy so that it can initiate the required signalling for inbound traffic. This overcomes the major weakness of current RSVP proxy Internet drafts. This minimises any signalling overhead. Further details are in annex A4.4.10.

The location of the proxy needs to be carefully considered. Where edge or bandwidth broker admission control is used within the BAN, the natural location of the proxy is at the BAR. Where hop-by-hop call admission is used through the BAR, the proxy needs to sit at the exit from the BAN – at a gateway node. However, the inbound and outbound gateway nodes are not necessarily the same node. Thus, the outbound gateway node could have the QoS information that is actually needed by a different node. One mechanism to avoid this problem is to have proxy manager functionality. This (which may be a centralised manger or distributed between all gateways) allows the nodes to discover the true inbound gateway of a "missing" data stream. This aspect of the solution is not scalable, but may work, as the number of gateway nodes is usually very restricted, and the probability of needing to call upon this functionality is small. This problem does not exist if there are restrictions on the data route through the network, as is the case in the BCMP. Here, the anchor nodes are always fixed for in and out-bound traffic, thus if these nodes act as RSVP proxy nodes, the inbound and outbound data route through the proxy is guaranteed. The scheme also allows RSVP to be used to signal DiffServ Code Points in the BAN using the RSVP DCLASS object. The mobile node can use the DCLASS object to instruct the proxy node to mark incoming traffic with certain DiffServ Code Points to trigger different forwarding behavior within the access network. Thus the mechanism can also be used to give relative priority to specific incoming flows, without explicit resource reservations. It can be considered a means to update a service level agreement.

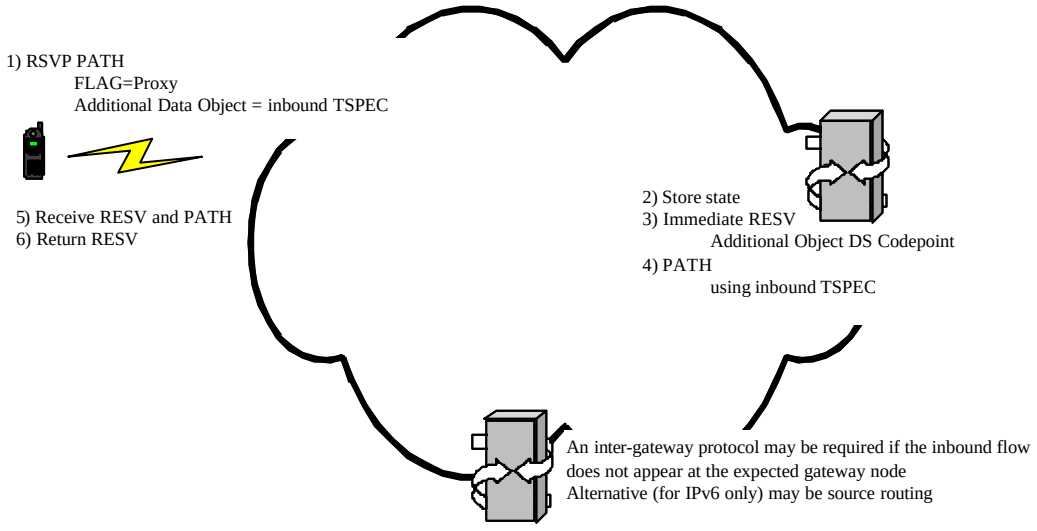


Figure 4-5: Use of RSVP Proxies

4.4 Extending the Base-line architecture

QoS can be provided in a mobile, wireless Internet using the Internet standard ISSLL architecture. Key elements of this architecture are already being seen in both terminals (as an example, Microsoft implement RSVP) and networks (as an example, BT uses DS routers). However, to achieve the best performance from the system, enhancements are required. The choice of suitable enhancements is dependent upon the operating environment. However, here we attempt to highlight some ways in which the different extensions might be used. In many cases, extensions are independent. Where there are interactions, these have been highlighted.

4.4.1 The strengths of service guarantees that can be achieved are limited because of the call admission architecture

The edge nodes can only make approximate admission control decisions. In an access network with either little capacity for statistical multiplexing effects or restricted bandwidth, the quality of these decisions will be weak. The possibility of mobility further weakens the strength of such decisions. The standard IntServ classes are not well suited to the wireless environment

Solution: Either Bandwidth Broker or Hop-by-hop Admission

Bounded Delay Service

Simplified service definitions and signalling protocols

Mobility Enhanced QoS parameters

The use of the bounded delay service with hop-by-hop admission gives good QoS guarantees without many of the scalability problems normally associated with hop-by-hop solutions. In particular, per flow state needs only to be maintained at the edge of the network, as within a high capacity backbone network, nodes need only maintain a bandwidth sum. This approach may be of particular importance to an operator with restricted fixed network bandwidth. The bandwidth broker solution, which could also gain benefit from the BD service, may be more appropriate to an operator with less bandwidth restrictions, that does not want the overhead of replacing existing routers.

The use of simpler QoS classes within the end-to-end RSVP messages needs to be treated with caution, and are best supported only after standardisation has occurred. Interworking functions must be provided at the domain edge to ensure inter-operability with legacy IntServ networks, and unless rapid standardisation occurs, these legacy systems are likely to predominate. Simpler QoS classes and mobility enhanced QoS parameters can easily be considered as part of a local BAN signalling solution.

4.4.2 Minimize the quantity of signalling required

Standard ISSLL RSVP operates in a soft state node, with refresh messages every 30s. These messages are large. Mobility may even trigger the mobile node to send more RSVP messages to repair a broken path.

Solution: Hard State RSVP

Context Transfer protocol

Simplified service definitions and signalling protocols

The use of RSVP in hard state mode is recommended, at least across the wireless interface. The context transfer protocol is also valuable in reducing the signalling across the wireless interface. It means that the MN need not participate in further admission control signalling no matter how much the network path is disrupted.

The use of simpler QoS classes within the end-to-end RSVP messages needs to be treated with caution, and are best supported only after standardisation has occurred. Interworking functions must be provided at the domain edge to ensure inter-operability with legacy IntServ networks, and unless rapid standardisation occurs, these legacy systems are likely to predominate. Simpler QoS classes can easily be considered as part of a local BAN signalling solution

4.4.3 Seamless handover

This can be considered as having several elements. Fast creation of reservations over the wireless link. Fast creation of reservations within the network. Handling of traffic whilst the handover and reservation creation is taking place – this is important if reservations are not established before handover takes place.

Solution: Context Transfer protocol

Repaired RSVP local Path repair

DS Handover markings

Mobility Enhanced QoS parameters

Reservations for temporary tunnels

The context transfer protocol will enable reservations to be established quickly over the wireless link, in such a way that the new router will understand how to map the IP flows into the new layer 2 reservations.

These link layer reservations may be used for network layer prioritisation traffic as well as reservation based traffic.

Repaired RSVP local path repair, which relies on the context transfer protocol, and DS handover markings are recommended where hop-by-hop call admission is used. Local path repair is not required when a bandwidth broker or edge admission control scheme is used, as there is no path to repair. DS handover markings are impossible to implement unless nodes have per-flow state, so this cannot be used with bandwidth broker or edge based call admission. In these cases it is unlikely that the admission control decision will be based on sufficient information to make a minor path re-route significant.

Unless the BAN is over-provisioned, a solution to provide some QoS to packets re-directed in flight is required. The DS handover marking approach is the simplest and the only solution when unplanned handover takes place. Firmer QoS support is given when QoS reservations for temporary tunnels are used, but this has an increased complexity and signalling overhead.

4.4.4 Reservation QoS in the absence of end-to-end QoS signalling

This requirement is of particular interest if the access network is bandwidth limited.

Solution: Local BAN signalling Protocol

RSVP proxies

If the BAN is limited in resources or the operator wants to provide a wider set of QoS services, local signalling or RSVP proxies can be added to the infrastructure. These mechanisms provide local reservations, enabling the MN to obtain better service, especially from the wireless link or bottleneck local network.

In many cases, applications are already either explicitly or transparently using proxies. For example web proxies enhance data transfer by providing data caching, may modify data to suit the capabilities of the terminals, and improve the service provided by protecting the user from unsavoury sites. Even if reservations existed only between such proxies and the MN, the service to the MN could still be much improved, providing a commercial advantage to an operator.

RSVP proxies will give an adequate solution to this problem. However, if end-to-end QoS is rarely available then the local BAN protocol approach may be able to give a better, more complete, solution. Both solutions require that the MN can identify when it wants to request local rather than end-to-end reservations. However, this could be automated so that the user is not actively involved in this process.

4.4.5 Discussions

From the above, it is worth highlighting a few key points. The first is that a local BAN signalling protocol could be developed which solved, using the same mechanisms as the stand-alone extensions, all of the problems within the BAN. This approach, which must be handled carefully to prevent developing a network specific solution, needs to be further investigated.

The second key point is that many of the extensions rely for best performance on the existence of the context transfer protocol. This protocol is being developed within the IETF Seamoby working group. Its use, once developed is recommended.

IP Tunnelling is a highly contentious issue within the IP community. The use of IP tunnels can interact negatively with QoS. Whilst this is really an issue for the tunnelling mechanisms rather than the QoS mechanism, tunnelling has nevertheless been considered in the development of this QoS solution. Further discussion can be found in Annex A4.5.4

Finally, some of the extensions do not yet exist. Many of the extensions have been specifically developed or refined by the BRAIN team. It is intended to implement many of these extensions to further test the ideas. Most importantly, we intend to submit these to the relevant IETF working groups to ensure fast standardisation. This will prevent “BRAIN specific” stovepipe solutions and lead to full integration with the Internet. Thus we can say that the QoS solution developed sits within the IETF ISSLL framework, with totally standard interfaces. It can be truly said to be a true IP, end-to-end solution.

5 BRAIN Interfaces

5.1 Introduction

The following sections give an overview of the BRAIN Interfaces. It is generally considered that providing a clear separation between functional elements part of each layer in a protocol stack is essential for an open architecture. As depicted in Figure 5-1, three interfaces are introduced. The Enhanced Socket Interface (ESI) and Local Management Interface (LMI) are introduced to make the application independent from the functionality of lower layers. To get a clear distinction between the network and link layers the IP2W is introduced, making it possible to implement BRAIN access networks using various wireless technologies as long as they support some general BRAIN requirement in term of services provided to the network layer.

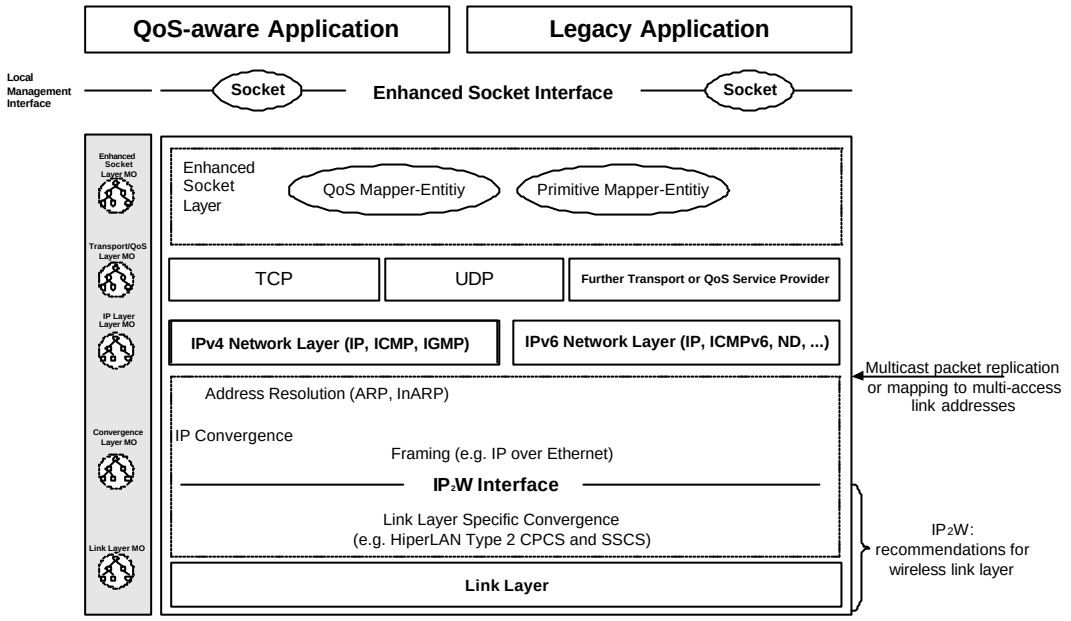


Figure 5-1: Brain Mobile Terminal Stack

The ESI formalizes the way applications control end-to-end QoS. It is a generic interface meaning it is independent of any platform[3], supported QoS and any transport service provider. The Enhanced Service Layer supporting the ESI functionality by the means of Mapper. The LMI can be seen as a complementary to the ESI. It provides a set of functions, local to the terminal for monitoring and controlling local resources. IP2W provides a unified interface for controlling various capabilities of wireless interfaces and makes useful information from the link layer available to upper layers. As depicted in Figure 5-2 the ESI and LMI are available on the mobile terminal and correspondent host whereas the IP2W Interface is supported on the mobile terminal and the BAR. The following section contains a rough explanation about each interface. More detailed information can be found in the annex A5,A6.

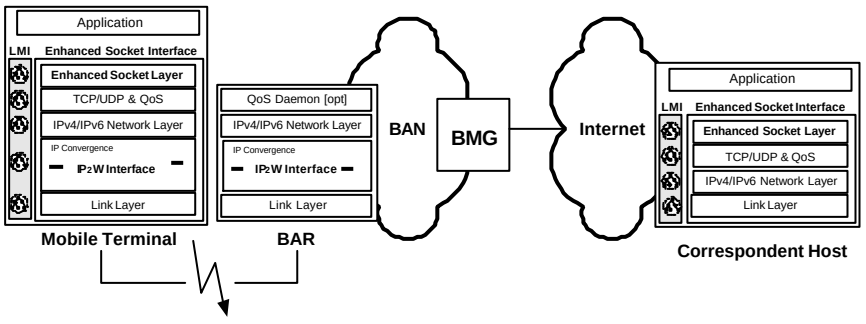


Figure 5-2: Availability of ESI, LMI and IP2W

[3] In contrast to Microsoft GQoS which can only be used with Microsoft's operating systems

5.2 Enhanced Socket Interface

This chapter addresses the Enhanced Socket Interface, the concepts behind it and their relation to the QoS Architecture introduced in section 4.

5.2.1 Design Principles

A set of overall design principles was agreed that guided in the design of the interface. The following items summarise the design principles applied to the ESI and ESL.

[DP 1] The ESI is an extension to a non-QoS aware transport service interface. It extends the ubiquitous used transport service interface by QoS primitives.

[DP 2] The ESI is a generic interface, which means it is independent of any platform, supported QoS-, Network- and Transport Service Provider.

[DP 3] The ESI makes the development of QoS aware application possible and it supports non-QoS aware applications.

[DP 4] The ESI considers only *end-to-end* Quality of Service means between a Sender and a Receiver. All primitives are therefore *end-to-end* QoS related primitives.

[DP 5] The ESI does not introduce or enhance any existing QoS protocols - the semantic of the primitives must be realised by the available QoS Service Provider. There is no additional signalling introduced beside that of the used QoS Service Provider. Note, a QoS negotiation protocol is introduced in BRAIN Work Package 1.

[DP 6] It is assumed that there is a pre-configured protocol-stack with a pre-configured ESL, a Connection-oriented -, Connection-less - and at least one QoS Service Provider. The set up can be subject to the mobile user's contract with the network operator. The facility to change the default settings, especially the default used QoS- and Network Service Provider is out of scope of the ESI/ ESL. (See [DP 7])

[DP 7] Local Management Issues like information about available QoS, Transport or Network Service Providers are not considered in the ESI. Their functionality is part of the Local Management Functionality and can be access through the Local Management Interface.

5.2.2 Design Decisions

The ESI supports QoS aware application with a very generic interface. Due to the independence from the used QoS Service Provider no detailed information can be offered to the upper layer. The following main characteristics serve as the basis for the ESI and ESL

[DD 1] A confirmed service, supporting the upper layer with information about whether the requested QoS can be supported or not. Therefore an appropriate explicit end-to-end signalling protocol must be available.

[DD 2] An unconfirmed service, which does not support the upper layer with information about whether the requested QoS can be supported or not. No additional explicit end-to-end signalling protocol is required, which enables sending of information immediately.

[DD 3] Notification Service, indicating the violation of a QoS aware flow.

5.2.3 ESI Primitives

Based on the Design Decisions and Design Principles the following services are supported by the ESL through the ESI.

SetQoS Service

That confirmed service acknowledges whether the requested QoS for a specific flow can be granted or not. The confirmation will be a positive acknowledge if the end-to-end QoS can be granted or a negative one if not. Due the above defined design principles [DP 5] this service can only be offered if the protocol stack provides an appropriate end-to-end signalling capable QoS Service Provider, if not this kind of service can not be supported to the upper layer.

ChangeQoS Service

After establishing a QoS aware flow with the *SetQoS Service* the QoS requirements of the receiver or sender may change. This leads to a change of the reserved resource between the sender and receiver (inclusive). This service can only be applied to already established QoS aware flows, set up with *SetQoS Service*. It is up to the currently available QoS Service Provider how the change from the old, already guaranteed QoS, to the new required QoS can be accomplished. This service can only be offered if an appropriate explicit end-to-end signalling QoS Service Provider is available.

AssignQoS Service

AssignQoS offers the service of sending packets with a certain assigned QoS class. There is no explicit end-to-end signalling involved meaning that there is no overhead in setting up a QoS aware flow and doing reservation. If no such QoS SP is available this service cannot be supported.

Violations Services

Violations of QoS might happen due to admission or policy control, handover or simply because the network entity cannot grant the requested QoS. Different types of violations have to be distinguished:

SetQoSViolation is triggered if a violation happens during setting up a QoS aware flow – means during the usage of the *SetQoS Service*.

ChangeQoSViolation is triggered if a violation happens during changing the QoS properties of a QoS aware flow - means during the usage of the *ChangeQoS Service*.

QoSViolation is triggered if a violation happens not during the usage of *SetQoS* or *ChangeQoS Service*

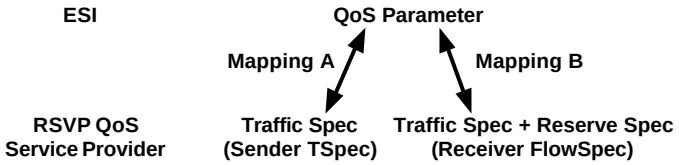
ReleaseQoS Service

ReleaseQoS can be used as an unconfirmed service mapped to the QoS Service Provider to tear down specific primitives.

The service's primitives have in common that they use mostly the same signature comprising a parameter *flow* - descriptor of the flow to be handled - and a parameter *QoS* - the QoS Parameter.

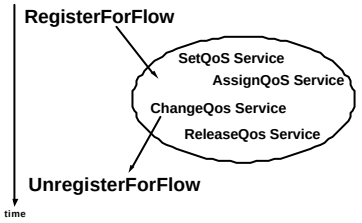
5.2.4 QoS and Primitive Mapper

The Enhanced Socket Layer (ESL) provides with the functionality needed by the ESI. It consists mainly of a QoS Mapper and Primitive Mapper-Entity, which map the ESI functionality to the available transport and/or QoS Service Providers functionality. It's up to the terminal implementation to decide how the Mappers should do this. The mapping is influenced by several factors mainly reflecting the QoS Service Provider's nature. Assuming the mobile terminal is equipped with an RSVP-like QoS Service Provider, then as depicted in the figure on the right, the mapping can be subject to the ESI's QoS Parameter: service type. If an application requests a service type Guaranteed the ESI's QoS Parameter has to be mapped according to Mapping B. All other requested service types can be mapped with Mapping A. The benefit of using Mappers is that if at some point the QoS Provider is exchanged e.g. due to handover then simply the mapping has to be adapted. The application is not affected by the change. More details are in annex A5.11



5.2.5 Legacy Application Support

A Legacy Application is one that is QoS unaware, meaning it cannot participate in establishing end-to-end QoS. However it is possible that a user would like to use a legacy application on its mobile terminal – but have it QoS enhanced. The basic idea to do this is that a third party application on the terminal takes over and manages the QoS for the Legacy Application. An application doing that is called a Configurator. The details of how a Configurator works are implementation specific, but in general it can be supported either by the means of a) the Local Management Functionality or b) additional primitives offered by the ESI. In the case of using the Local Management Functionality (LMF) the Configurator can access and modify the traffic control properties like packet classifier, admission control and packet scheduler. Complementary to the LMF the ESI optionally supports additional primitives enabling the Configurator to get access to the Legacy Application's flows. QoS aware applications can use an ESI primitive like *QoSSocket* to get a flow description. However, the Configurator cannot, since it is not an end-point for these flows. Therefore it needs some other kind of primitives, to enable it to 'read' the flow descriptors that are required for use of the ESI's services. As depicted in the figure on the right two additional services are introduced - *RegisterForFlow* and *UnregisterForFlow*. *RegisterForFlow* gets information about flow descriptors for specific flows. These flow descriptors are necessary to associate QoS with these flows to enable the usage of the ESI's services.



5.2.6 Local Management Functionality

Future broadband wireless multimedia applications shall be able to run on a set of different terminals over a variety of different networks. In order to be aware of the terminal capabilities and to manipulate the operation of the terminal, applications usually interact with the operating system to discover available network adapters, the state of the network and other required features. Furthermore, the operating system usually provides control functions that allow fine grain control over the behaviour of the terminal. A set of mandatory and optional management functions has been identified that are useful to support specialized multimedia applications. In order to avoid operating system specific functions, local management functions in terms of an abstract object model are defined. The advantages of this approach are two-folded. First, it allows to concentrate on the management functions itself, without being distracted by the way, a specific operating system might implement this. Second, it allows mapping this specification to an API that abstracts from OS specific functions. This layer between the OS and the applications enhances the portability of applications.

5.2.7 Multi-homing

BRAIN terminals might have several network interfaces to different networking technologies, for example HIPERLAN/2 and UMTS. In this scenario, a terminal might be connected simultaneously to several networks, i.e. multi-homed.

In a multi-homed scenario, local management functions report the current status of the different network interfaces. The protocol stack then decides upon performing handover for existing connections. For new communication activities, the multi-homed scenario offers the ability to select which network to use by default. A further, often overlooked, aspect is that a user might have different subscriptions with different network operators (see the no coupling and loose coupling scenarios in the BRAIN-UMTS, section 2.4.2) and might want to very selectively define which network service to use for which communication application. BRAIN therefore defines local management functions that can be implemented by a sophisticated terminal to:

?? select the default network to be used for further communication activities,

?? control the handover in the multi-homed scenario,

?? allow selective assignment of IP data flows to certain network interfaces and, if required and provided by the protocol stack, to certain IP gateways.

5.3 IP to Wireless Interface

This section introduces the IP to Wireless Convergence Interface (IP_2W). The following goals and principles have been applied in the design of the link layer interface:

?? Provide a unified interface for controlling the various capabilities of wireless interfaces

?? Make useful information from the link layer available to upper layers

?? Separate handover control and resource management clearly into link layer internal functions and functions controlled by the network layer

?? Produce guidelines for implementing "IP friendly" wireless link layers

?? Discourage policy decisions in the link layer

?? Preserve layer transparency

The above principles lead to the design of a link-layer service interface that gives upper layers more control over a wireless link layer, while leaving room for different implementations. A central design issue in the IP_2W interface is how to divide the responsibilities between the link layer and the upper layers. For instance, QoS, handover, and idle mode have an impact on multiple layers of the protocol stack. The IP_2W interface makes the boundary between layers explicit. Flexibility is retained, by making some of the capabilities optional. An IP_2W compliant link layer advertises the capabilities it supports through a generic configuration management function, allowing the higher layers to adjust.

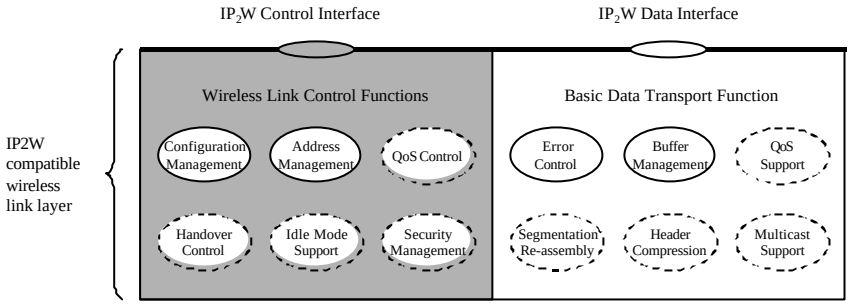


Figure 5-3: IP$_2$W Interface Model

Figure 5-3 presents the interfaces and functions defined in the reference model. The IP$_2$W interface is separated into a Control Interface and a Data Interface (i.e., a separation between user plane and control plane is identified). Each interface offers access to a set of functionality on the link layer. Several distinct functions have been identified, represented by the small ovals under the interfaces. The dashed ovals represent optional functions, which might not be supported by all wireless links.

The *Data Interface* deals with sending and receiving user data (i.e. the user plane). Each user data packet passed from the network layer to the link layer is accompanied with an auxiliary *Interface Control Information* (ICI) block. The ICI gives the link layer additional information on how to deal with the packet and where to send it, such as identifying the QoS and security context that should be used on the link layer. In addition, several user plane procedures, typically found in a wireless link layer, have been identified. The procedures are technology dependent and are not specified as part of IP$_2$W but their impact on the performance and QoS of IP traffic in different situations has been investigated.

Each functional block of the *Control Interface* corresponds to a set of service primitives that can be used to configure and control specific aspects of the link layer operation:

?? *Configuration Management* function is a general management function for discovering and configuring the capabilities of the link layer.

?? The *Address Management* function deals with the assignment of hardware addresses and IP addresses to network interfaces, and maintaining mappings between the two.

?? The QoS Control function deals with quality-of-service support on the link layer, including support for both reservationless QoS (e.g. priorities) and reservation based QoS (e.g. bandwidth reservation).

?? The Handover Control function consists of several subfunctions that allow reliable movement detection, tracking the current handover phase, and control of handover timing and access point selection.

?? The Idle Mode Support defines a link layer interface for allowing a mobile node to put the wireless link layer into a standby mode in order to save radio resources and power.

?? The Security Management function provides a way to enable and configure keys for link layer authentication and encryption, if supported by the link layer.

The following sections highlight some of the more novel aspects of the IP$_2$W control interface. A more detailed discussion and the complete specification can be found in annex A6.

5.3.1 Handover Support

The generic handover interface in IP$_2$W provides a high level abstraction and service primitives for the handover control procedures required in the different types of handover. The central design requirement is that the interface must be independent of specific wireless link layer technologies, focusing instead on the services needed by the network layer. The IP$_2$W handover control interface consists of three major functions: *neighbourhood awareness*, *handover progress monitoring*, and *handover decision control*. The following sections discuss how these basic functions are used to perform a handover within a BAN.

5.3.1.1 Movement Detection

IP mobility protocols have traditionally tried to make minimal assumptions about the link layer between the MN and an access network. Typically, mobile-controlled handover is the only supported handover type and access router discovery and movement detection are based purely on network layer mechanisms. However, it has been noticed that the anticipation of a prospective handover to a new router is essential in

achieving fast and smooth handovers. For instance, movement detection based on network layer mechanisms (e.g. as used by Mobile IP) is not compatible with the maintenance of QoS for services that expect short delays and/or low packet loss. Recognising this, recent handover proposals in IETF assume that link layers can provide "triggers" that can accelerate the IP level handover procedures.

IP_2W defines a mechanism that bases movement detection on making link layer handover control information available to the upper layers. Movement detection in IP_2W is based on the generic *neighbourhood awareness* and *handover progress monitoring* functions of the IP_2W handover control interface. To support this functionality, the link layer must be capable of independently monitoring the radio link quality and initiating measurements on neighbouring radio transmitters. As a technology independent interface, IP_2W does not specify, how the link layer should collect the measurements. It merely states that the information should be available in the MN at the time of handover, in the format defined by the IP_2W interface.

Handover progress monitoring allows monitoring of current handover phase (in a BAR, for each attaching or detaching MN), BAR selection, and handover timing. The handover phases can be identified as *preparation, decision,* and *execution*. The IP_2W signals the upper layers of the handover phase via an event notification. In the MN, handover progress notifications have a direct relationship with movement detection. In the BAR, handover progress events may be used for triggering fast and smooth handover mechanisms on the upper layers.

A link layer that supports the neighbourhood awareness control function gives the network layer a list of nearby BARs that are good candidates for a handover. The list contains the identifications of candidate BARs. The BARs may be identified, for example, by their hardware addresses, IP addresses or NAIs. This list is passed to the upper layers through an event notification. The network layer can use this information to anticipate, where the MN is about to move next.

5.3.1.2 Planned Handover

Planned handovers are possible if *handover decision control* in addition to neighbourhood awareness is supported by the link layer. In a planned handover, both the network layer resource management and the radio resource management on the link layer take part in the handover process and the selection of the new BAR.

Neighbourhood awareness provides the MN a view of nearby BARs that are candidates for handover, ranked in the order of preference. Each entry is accompanied with a comparison value indicating the "goodness" of a particular BAR. It is assumed that the link layer on the MN will calculate the value, for instance, based on RSSI (Radio Signal Strength Indicator) measurements and information transmitted by base stations. The information provided by neighbourhood awareness depends on the internal procedures of the link layer and may be unavailable except at well defined times. It is assumed that the information is complete and available at least when the link layer is ready to transition from the handover decision phase to the handover execution phase.

Handover decision control allows control over the BAR selection and over the exact timing of the handover phases. For controlling BAR selection, the information provided by neighbourhood awareness is required. For controlling the handover timing, handover progress monitoring is also required. When the handover progress monitoring function signals that the previous handover phase has been completed, the handover decision control function can be invoked to allow the handover to proceed to the next phase.

Figure 5-4 gives an example of handover signalling in a planned handover scenario where the MN detaches from the old BAR before connecting to the new BAR. The message sequence chart shows how handover event notifications at the IP_2W interface can facilitate proactive IP handover preparation between the access routers and how they can trigger the registration procedure at the new BAR.

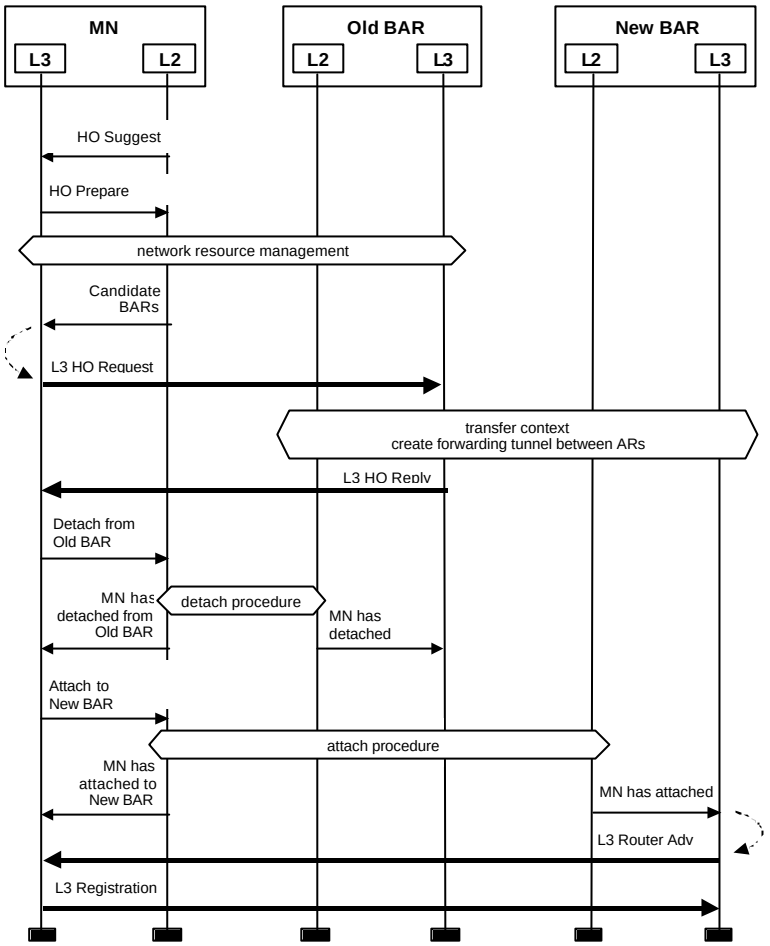


Figure 5-4: Planned Handover

The handover begins with a notification from the link layer (HO suggest) that the link quality has degraded enough to warrant a handover. The handover control in the MN then orders the link layer to start scanning for BARs that are good candidates for handover (HO Prepare). While still connected to the network through the old BAR, the MN performs network resource management procedures and finds out which neighbouring BARs have sufficient resources to support the MN. When the link layer is ready for handover (Candidate BARs signal), the MN combines the link layer and network layer views and selects the new BAR. The MN then notifies the old BAR of the imminent handover, possibly triggering a context transfer protocol between the old and new BARs. After receiving an acknowledgement to the handover request, the MN detaches from the old BAR and attaches to the new BAR, followed by registration and authentication procedures on the network layer.

5.3.2 Idle Mode Support

Section 3.5.1 identified two separate but interrelated concepts of state with respect to the mode of activity, which can be coarsely characterised as follows:

?? Active/idle/detached according to IP packet transmission activity

?? Active/standby according to (link layer) power management

Hence, it may be noted that idle mode saves radio spectrum and routing state in the BAN whereas standby mode saves battery in the MN. These two concepts may coincide in a MN but not necessarily; an idle MN needs not be in standby mode even if power management is supported.

The link layer standby mode support for BAR includes:

?? periodically broadcasting a paging area identifier that allows MNs to recognise their current paging area

?? broadcasting a paging request to a stand-by MN (when requested by a network layer paging protocol)

The link layer standby mode support for MN includes:

?? supporting the means to switch between active and standby modes

?? monitoring paging area identifiers broadcast by BARs

?? monitoring paging requests broadcast by BARs while in the standby state

?? alerting the network layer when a paging area change or a paging request is detected and associating with a BAR in order to enter active mode

Performing paging over the wireless link at the network layer will unfortunately require maintaining link layer connectivity (or at least the ability to receive selected IP packets broadcast or multicast by the network layer). This does not allow the network interface of the MN to enter a standby mode. Therefore a paging scheme should use link layer signalling over the radio link, preferably over a broadcast or multicast paging channel. The MN gets all the distributed paging requests via that channel and determines if it is relevant for the terminal based on an identifier of the MN included in the paging packet parameters. The paging identifier that the MN recognises can be configured through the IP_2W interface. Having received a valid paging request the MN wakes up through internal mechanisms and sends a location update using network layer signalling.

To illustrate Figure 5-5 shows the paging request at the MN.[4]

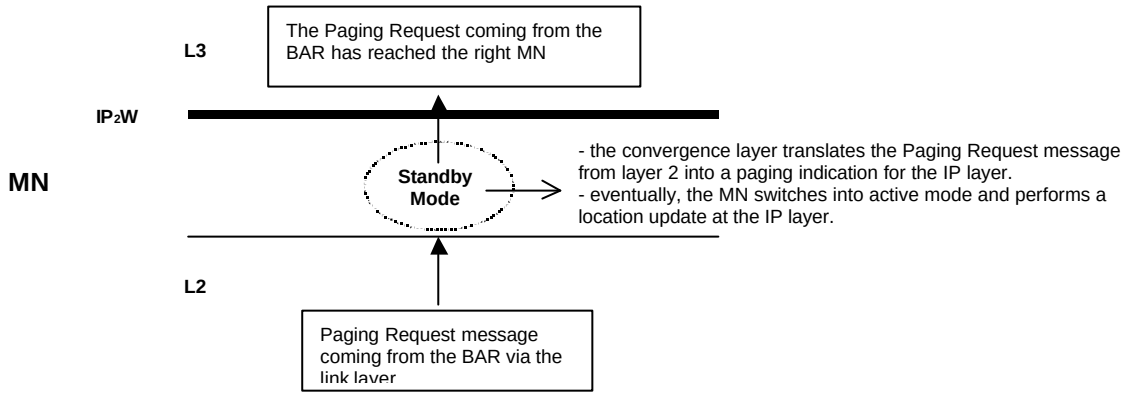


Figure 5-5: Paging Request at the Mobile Nodes

5.3.3 QoS Support

Traditionally, IP architectures designed to improve the QoS given to various flows are based purely on IP-layer decision-making, packet buffering and scheduling. Few assumptions are made about the link-layer. However, some new link layers offer more sophisticated functionality than simple first-in-first-out packet delivery. For example, the HIPERLAN/2 link layer can support priority-based packet scheduling as well as guaranteed bandwidths to individual flows.

Since the link layer has a better understanding of the status of the communication medium, radio equipment manufacturers tend to try to implement as much of the packet forwarding decisions as possible below the IP layer. However, IP-based protocols are increasing in numbers and complexity, thus, implementing too much intelligence in the lower layers is likely to increase implementation complexity and introduce drastic layering violations; it can be argued that IP protocols are best handled within the IP layer. It seems apparent that a solution between these two extremes is needed, a generic convergence interface between the IP QoS protocols and the link layer mechanisms.

In the IP_2W model the link layer provides a certain number of QoS contexts. These contexts are established in such a way that they can be used by both a Differentiated Services (DiffServ) or by an Integrated Services (IntServ) flow.

5.3.3.1 Connection-oriented QoS based on specific flows

Performing QoS mapping on a flow basis (e.g., IntServ) requires setting up a link and reserving bandwidth before data transmission can take place. The bandwidth reservation is initiated by, e.g., a

[4] NB The interface is asymmetric: the MN and the BAR see different primitives and messages crossing the interface.

RSVP procedure taking into account the traffic characteristics. After accepting the new connection (flow) a specific flow identifier needs to be established for that flow between the link layer and the IP layers. This allows distinguishing the specific flow from other flows. If the link layer does not provide explicit reservations of resources, other measures need to be taken, for example, specific scheduling at the IP layer.

IP_2W defines primitives for reserving flows over the underlying link. The network layer reservation request can be mapped to the respective IP_2W primitive (either in MN or in BAR, depending on flow direction). The link layer independently performs the reservation signalling and admission control for the flow using link-layer internal signalling procedures. In a PMP (point-to-multipoint) access topology, the access point typically has the necessary information required to check the available resources and to handle the request.

5.3.3.2 Connection-less QoS based on packets with QoS parameters

Flows may also have more abstract QoS needs than strict bandwidths, for example, a relative priority and reliability of the packet transfer; the IETF Differentiated Services defines such relative priorities, which are implemented with specific Per-Hop Behaviours that define the treatment of packets on a hop-by-hop basis.

Before the transfer can start, the IP layer must request a flow identifier for a prioritised flow or a flow with more requirements than a default service. The identifier defines the mapping between specific QoS parameters set on the IP layer and those supported by the link. On the link layer the packets are mapped to the respective queue and scheduled. More complexity arises since each packet may belong to a different flow and therefore may have different QoS parameters, even though the relative priority would be the same. As a result, combinations of parameters need to be mapped to the respective QoS context on the link layer, if available.

IP_2W defines primitives for mapping prioritised flows to link layer QoS classes. Admission control is not supported for prioritised traffic and must be performed by the network layer (e.g. with the help of a bandwidth broker).

5.3.3.3 QoS mapping and scheduling

There are several QoS parameters defined within the context of DiffServ and IntServ/RSVP that have to be appropriately supported when finding the proper identifier given to the IP layer. It is suggested that a link layer should have a limited number of predefined contexts to support DiffServ classes (not all but a subset) and a best effort context, and a certain number of additional contexts to support the IntServ flows. The best effort context should be static (established just after the association phase) and the others will be dynamic (established when required).

The IP layer packet classifier adds an outgoing packet to a particular network layer queue and also assigns it to a particular link-layer QoS context. The associated QoS context identifier is added to the ICI of the packet allowing the link layer to subsequently decide how to handle the packet. The link layer can maintain a mapping table between QoS context identifiers and the respective queues. Packets sent through the data interface can then be assigned to queues based on the mapping.

To avoid congestion on the link layer, flow control is performed. The amount of data buffered on the link layer is a difficult optimization problem; there should be just enough data to keep the link layer scheduling fully utilized, but not more than that.

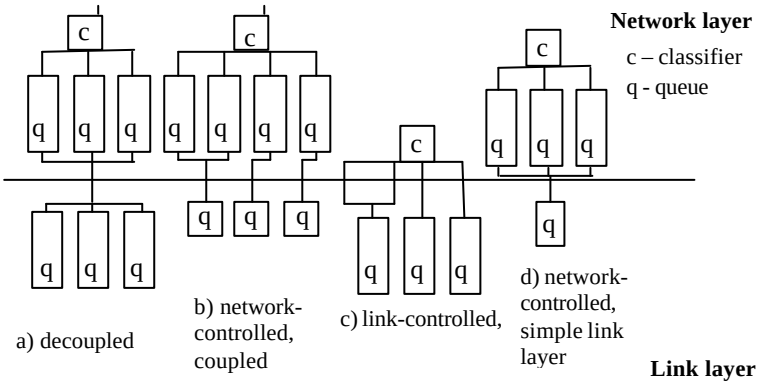


Figure 5-6: Various Scheduling Model Choices

Depending on the supported serving strategies and number of queues, scheduling is performed on the IP layer or the link layer. Figure 5-6 presents various alternatives to packet queuing and scheduling. In the case where the link layer is very simple and provides only one queue, scheduling takes part only on the IP layer (case d). If no scheduling is supported on the IP layer it might be handled solely on the link layer (option c). However, this requires that the IP layer can classify packets into several queues; otherwise a differentiated treatment of packets is compromised. In the case of HIPERLAN/2, scheduling is performed at least on the DLC layer with respect to the QoS contexts that have to be supported. Scheduling can be realized by a variety of mechanisms, including strict priority queuing, Weighted Fair Queuing (WFQ), Weighted Round Robin (WRR), or variants or Class-Based Queuing (CBQ). Similar mechanisms should be implemented on the IP layer. The interworking of the two separate scheduling mechanisms is a complex matter (case a). Layer violations should be avoided, for example, the link layer should not make its own decisions about the QoS needs of a flow, but rather get the exact need from the IP layer. However, leaving some scheduling to network and some to link layer can needlessly enlarge buffering and thus latency. The two schedulers could even work against each other if interoperability is not ensured.

The network-controlled, coupled model is deemed the most beneficial one. In this alternative, most of the queuing is done on the network layer (case b) and there is a clear division of responsibilities between link layer and network layer scheduling.

6 Worked example

6.1 Introduction

We return to the example scenario of section 1[5]: a Life in the Day of Carol, who uses a BRAIN "IP-based" network for all her services and applications. We identified several key network-layer elements, both protocols and an overall architecture, that need to be added to the current IP technologies to support her requirements: mobility management, QoS, security and interaction with higher and lower (link/radio) layers. Sections 2 to 5 have described our conclusions on these areas. Now, in the final section of the 'Core' part of this report, we outline how this work could be put together to deliver services to Carol.

Our purpose is to work through one example of how Carol's needs could be met, and thus illustrate that the full gamut of network-layer issues have been successfully tackled, and also to show that the output described in this Deliverable (D2.2) fits with that from the rest of the BRAIN project (D1.2 and D3.2). This 'worked example' also illustrates what we mean by an "all-IP" network and hints at some of trade-offs when IP design principles, such as layer transparency, are in tension with the desire for a modular architecture, for example.

It is also worth pointing out that there are several things that the example is **NOT** intended to do:

?? to help a BAN operator choose between various protocol options (here we just present one option, alternatives are in sections 2, 3, 4, 5 & the annex A1 – which is best will depend on the exact details of a particular scenario)

?? to explain our reasoning for choosing a particular solution (that's done in the earlier Sections, backed up by the various Annexes)

?? to be a detailed design specification (here we only do enough to demonstrate that this would be possible – and indeed will be done later within the MIND project for the MIND trial)

6.2 Registration

It is April 2004 and Carol starts her day at the University of Cheam, where she works as a physics lecturer. On arrival at the University she sits in one of the University's many cafes and starts up her laptop; this belongs to the University and was configured by the University support team – this is her home network. Her lap-top has a number of pre-loaded applications (Netscape, Word, NetMeeting, VideoPhone…), some BRAIN compliant middle-ware and a Windows 2002 stack. It is also equipped with a standard HIPERLAN/2 card with BRAIN compliant driver software loaded. The HIPERLAN/2 card looks immediately for beacon signals and, finding a University tagged HIPERLAN/2 signal it automatically begins to attach. A layer attach provides a link layer (MAC-id) identifier (220) and that is mapped at both the BAR and the MN to the static EUI-64 bit address assigned to the LAN card when it was manufactured.

The BAR acts as a Network Access Server (NAS box) and recognizes an attempt to connect at IP level and at the same time the OS on Carol's laptop is set to automatically begin login to the University network. Carol's laptop connects to the BAR and this, in turn, sends her details to the anchor point that "owns" the BAR. The anchor point makes use of AAA and DHCP servers to authenticate Carol and set up her laptop for networking. As part of the login Carol has to enter the number shown on a smart card she carries in her handbag (plus a 4 digit pin) – this number changes every 20seconds or so and is linked by a security association to the University AAA server. (Ownership of this Secure ID card is considered a suitable authentication). DHCP provides Carol's laptop with; an IP address (132.146.111.38), MTU, subnet mask, gateway address and DNS. The IP address belongs to one of the anchor points that control mobility management. In addition this action triggers the download of policy information, in particular QoS reservation access, from the policy server to the BAR resource management process.

The HIPERLAN/2 DLC layer provides L2 encryption over the air by generating keys at the beginning of the session. These are renewed after successful login.

[5] The exact details of the scenario presented here are slightly different; this is to allow us to draw out more clearly some key features of our work on the BRAIN network layer.

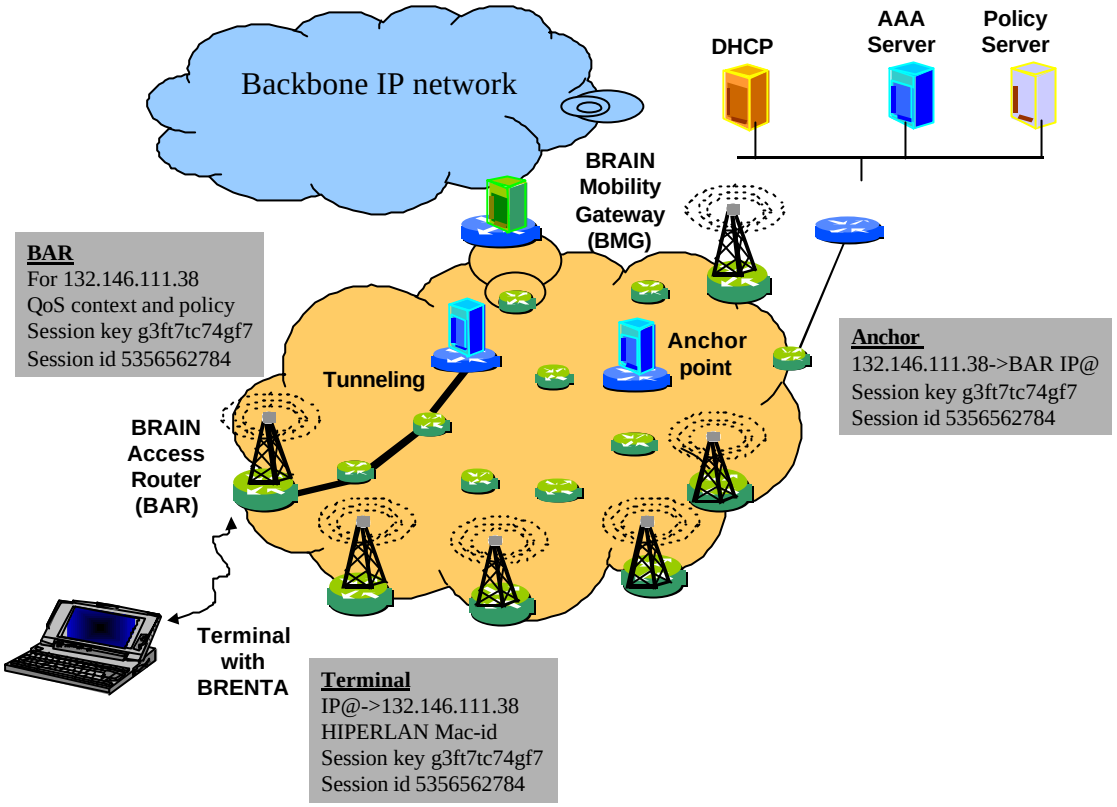


Figure 6-1: General Architecture of the University Network

6.3 Service example 1 - Multicast

Carol is now giving her first lecture of the day and each student has logged-in his/hers free laptops using the HIPERLAN/2 network within the lecture theatre. Carol wants to set up a video multicast session so that the students can see the details of a small experiment that she is performing. She starts her teaching applications package – written to take advantage of a SIP user agent and an Extended Socket Interface offering QoS support. This is a type B application in the BRENTA classification. Carol's SIP user agent has registered her as being on her laptop and using IP address 132.146.111.38. Carol's teaching application allows her to select the appropriate class (first year physics Mon 2pm), and sends an INVITE message to the class. This INVITE message contains a multicast address that Carol's application obtained from a pool. The INVITE is sent to the SIP proxy server and forked to all the members of the class at their current location – (this may include those that are not physically present in the lecture theatre). The INVITE message is sent via UDP and best effort since no QoS has been set up at this stage – the SIP UA takes care of reliable delivery. In this case it just so happens all the students are present in the lecture hall and that this is served by a single HIPERLAN/2 Access Point hop. Once the SIP INVITE procedure is complete, data transmission can take place. The application simply requires prioritisation QoS for the session, so the ESI.AssignQoS call is then used. The information in the traffic description is used by the OS/IP module to determine the network layer DSCP for the data, in this case the commonly understood DSCP for the Expedited Forwarding class. Carol's OS then makes a call to the IP2W interface, to ask for an appropriate tag for prioritised QoS over the HIPERLAN/2 radio link and then obtains a QoS context handle for the desired delay/priority class. Data can now be sent over this link. When the first packet arrives at the BAR, Carols SLA at the policy server is used to verify that she has permission to be sending EF data, and suitable policing functions are established to ensure that she does not overload the network. To ensure that the data is transmitted from the BAR to the students correctly, a multicast QoS enabled link needs to be established from BAR to the students. The BAR achieves this through mapping the IP multicast address to a link-layer multicast address, which it then used to forward the data packets over the wireless interface to the students.

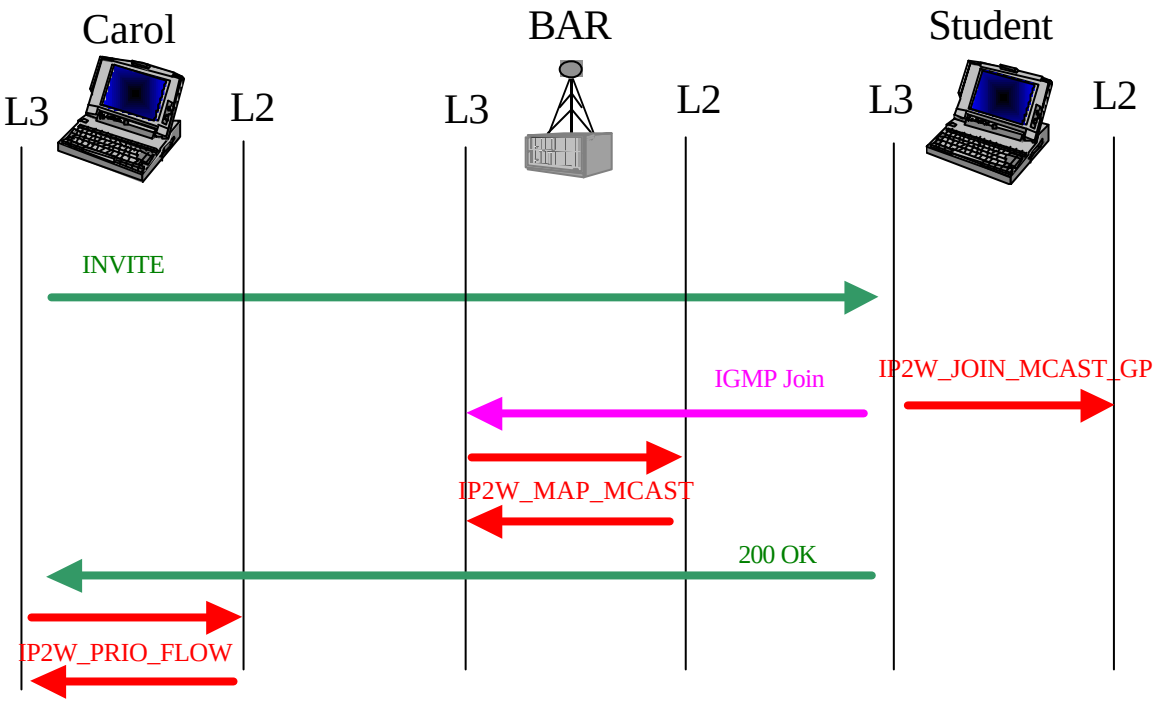


Figure 6-2: Message Flows for Multicast Example

6.4 Service example 2 - Voice

Carol finishes her class and goes back to the café to recover and tries to phone her stockbroker (a hot tip to sell Motorola, as a virus is infecting their phones). Carol's voice application, another type B application, attempts to contact John. Carol is using NetMeeting Plus and this is able to use DNS to resolve John's name to his permanent home IP address. NetMeeting then makes use of a SIP user agent to contact John and negotiate session parameters. Before "ringing" can take place both terminals must set up suitable QoS – this has been specified by Carol's SIP messages – she does not want a poor quality call.

Carol's application makes use of the SetQos.Request primitive of the ESI, this is mapped into an RSVP PATH message carrying the traffic description (TSPEC) which includes the maximum bandwidth of data that she wishes to send (64kbit/s). Also included in this PATH message is an ADSPEC object for the guaranteed service, which includes an estimate of the local processing and first hop delay. This will enable Johns application to accurately verify that Carols wireless connection will not overly degrade the quality of the conversation, The PATH message is interpreted at the BAR, where session details are recorded. Here also, the ADSPEC object is updated, as the BAR adds its estimate of the routing and transmission delay through the BAN to the total delay value. The PATH message travels all the way to John's terminal, being similarly processed at various RSVP-enabled nodes en-route, such as at the BMG. John's terminal replies with a RESV message. The BMG processes this as a standard RESV message, but the BAR node, as the entry to the network, takes additional responsibility for policing of the data, and so checks that Carol is authorised for this level of service, and a policing function is set up. Additionally, the BAR returns a DS Class object within the RESV message, giving details of the required DS packet marking to be performed by the MN. Similarly, John also sends a PATH message that reaches Carol who replies with her own RESV message. In this case, the BMG takes responsibility for the traffic policing functions. On receipt of the RESV message from John, Carol's OS make a request to the IP2W interface for guaranteed service for the specified FLOWSPEC. Similarly, the BAR on receipt of the RESV message from Carol, will set up the IP2W for the inbound data. In both cases the IP2W consults the Radio Resource Management function for admission control. In return the IP2W returns a tag for the packets and an indication of the incoming flow at the opposite end of the H/2 link.

After setting up the QoS Carol and John select encryption and this is provided by their terminals end-to-end.

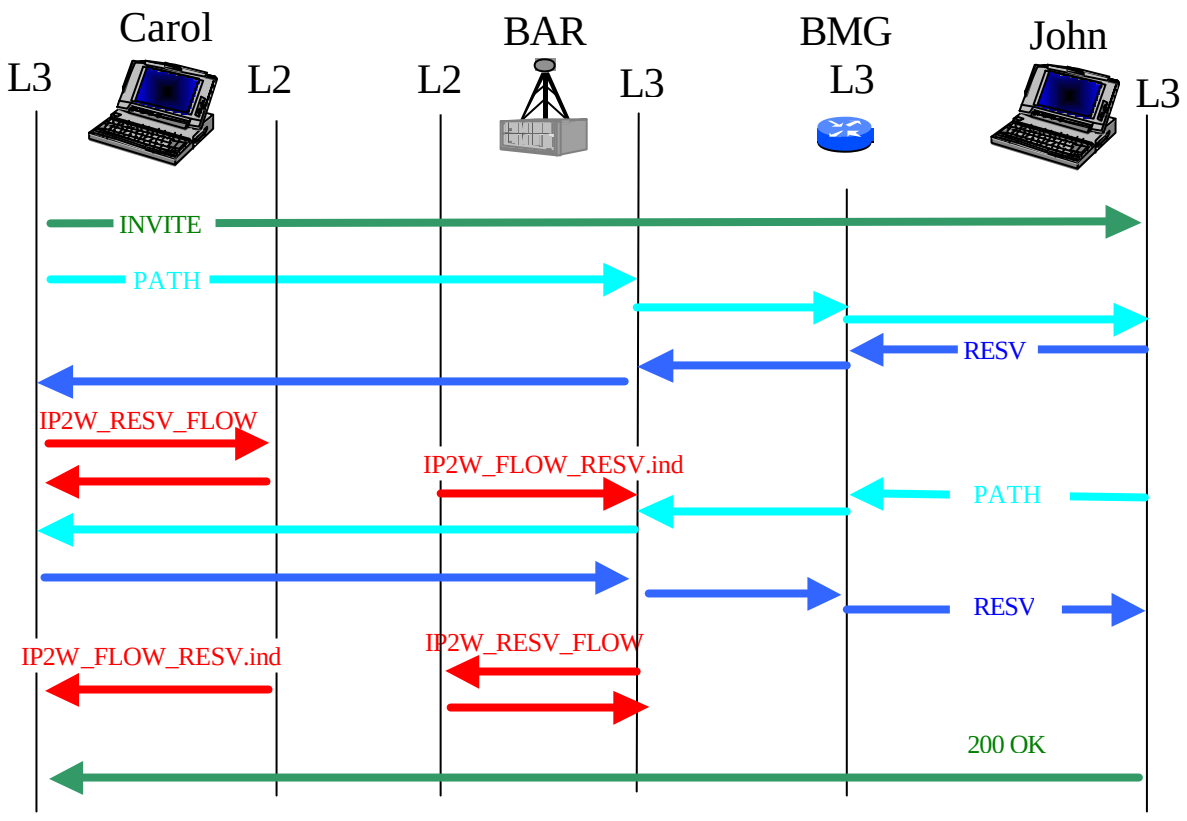


Figure 6-3: Message Flows for Voice Connection Across the BAN

6.5 Handover

Carol now moves from the café to the library as she continues to chat to John. Her L2 detects that the signal from her original BAR is waning and detects a much stronger signal from the library BAR. A Handover suggest is passed to MN L3 –the MN L3 sends a message to the RRM containing information on the radio strengths and ids of the BARs within range. This is followed by a solicitation for candidate access routers – the RRM and BAR conspire to return the candidates that can support the existing voice connection - In this case the library BAR is the only suitable candidate. The MN sends a handover request to the OAR and this, in turns, makes a handover request to the NAR, including: link layer address, IP address, session keys, and QoS context information. If this is successful a reply is sent back to the OAR that triggers the construction of a temporary tunnel for packet forwarding to the NAR. The MN is then responsible (MN controlled) for detaching from the OAR and attaching to the NAR. The MN has to do this quickly – it is operating a break before make – once the MN has detached from the OAR packets are sent down the tunnel to the NAR. As far as the packet flow is concerned the handover is not visible above the L2 – the same IP address is used and the handover reply has indicated the same flow id (6) when submitting packets to the IP2W interface.

The temporary tunnel can only be removed when the routing updates have been complete and the routing network can deliver IP packets carrying Carol's IP address to the NAR in a native way. In this case the University is running the BRAIN Candidate Mobility Protocol (BCMP) and the update from the NAR travels to the anchor that alters the tunnel end point for this address.

The NAR performs policing on the QoS flow – using the context transferred from the OAR. Within the BAN the University relies on its Gigabit Ethernet (over provision) and DiffServ to provide QoS. They have a monitor program that can contact the ingress node and act as a Bandwidth Broker to choke back QoS supported traffic at the ingress.

There is no need to execute any network QoS repair – since the BMG is unchanged and no bandwidth broker is used (the University rely on over-provision of the backbone).

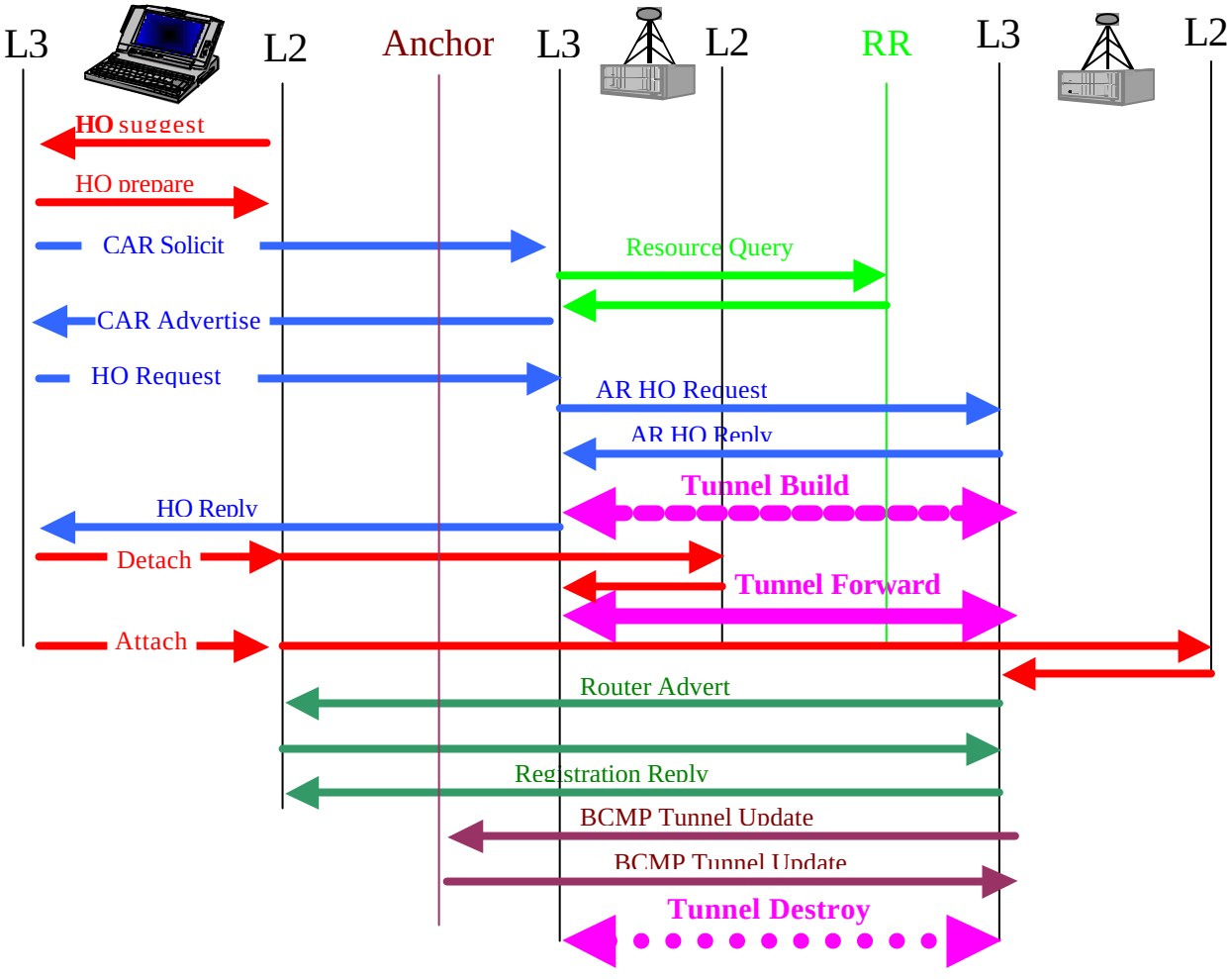


Figure 6-4: Handover Message Flow

6.6 Paging

Carol now finishes her conversation with John and sits down for a quiet read in the library. After 10 minutes of inactivity the IP layer informs the BAN that it wishes to go idle by sending an Idle Mode Request to the serving BAR. After receiving a reply from the network the IP layer instructs the HIPERLAN/2 driver to put her link layer in standby mode. So, the IP layer uses the IP2W interface to inform the HIPERLAN/2 drivers to listen on the paging channel for either a change of paging area identifier or a paging request for Carol's IP address (her IP address being given across the IP2W interface for this purpose). This involves powering down the transmitter and receiver and only listening to the paging channel during synchronized times

Carol picks up her sleeping laptop and decides to cross Railway Cuttings to the South Campus. On hearing the paging channel of a new BAR the HIPERLAN/2 stack compares the paging area identifier with the previously stored value. The university network administrators have divided the site into 2 paging areas – South and North Campuses and Carol has now moved across the paging boundary. The HIPERLAN/2 stack sends a L2 message (Paging Area Change indication) across the IP2W interface and causes Carol's mobility protocol to perform an unplanned handover from the BAR at the North Campus to the new BAR, which now becomes Carol's 'last serving BAR'. After the handover, the laptop re-enters idle mode. While Carol is wandering around South Campus, her laptop's link layer hears beacons from different BARs, but they all advertise the same paging area. Thus, Carol's laptop may continue its dogsleep.

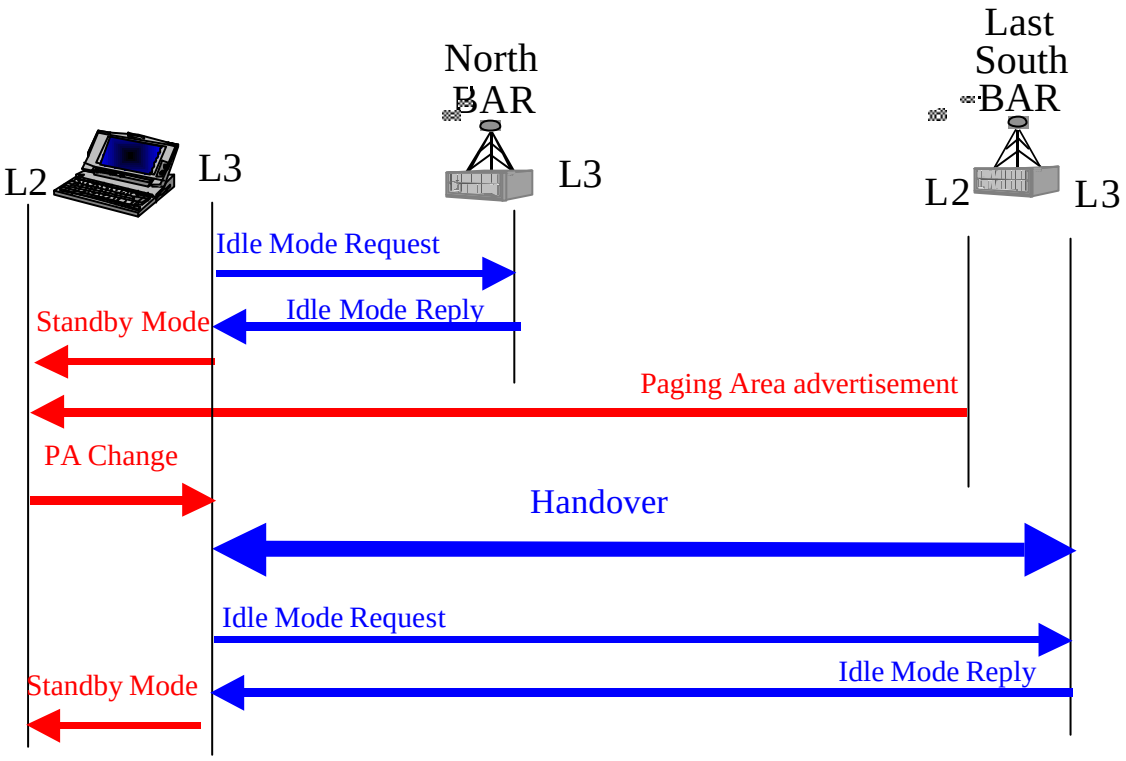


Figure 6-5: Paging Message Flow (Change of Paging Area)

When Peter attempts to contact Carol he sends packets to her IP address that he has cached from their previous session. When a packet reaches Carol's last serving BAR, the BAR has a mapping from the IP address to a multicast group consisting of all the BARs in the South paging area. The packets are buffered and a paging request is sent and, when it reaches Carol's layer 2 it recognizes the request as aimed at her and signals the IP layer to leave idle mode through the IP2W interface. The IP layer then executes a standard, non-planned, handover from the last serving BAR to the current BAR and the anchor updates its tunnel point end mapping for Carol's IP address.

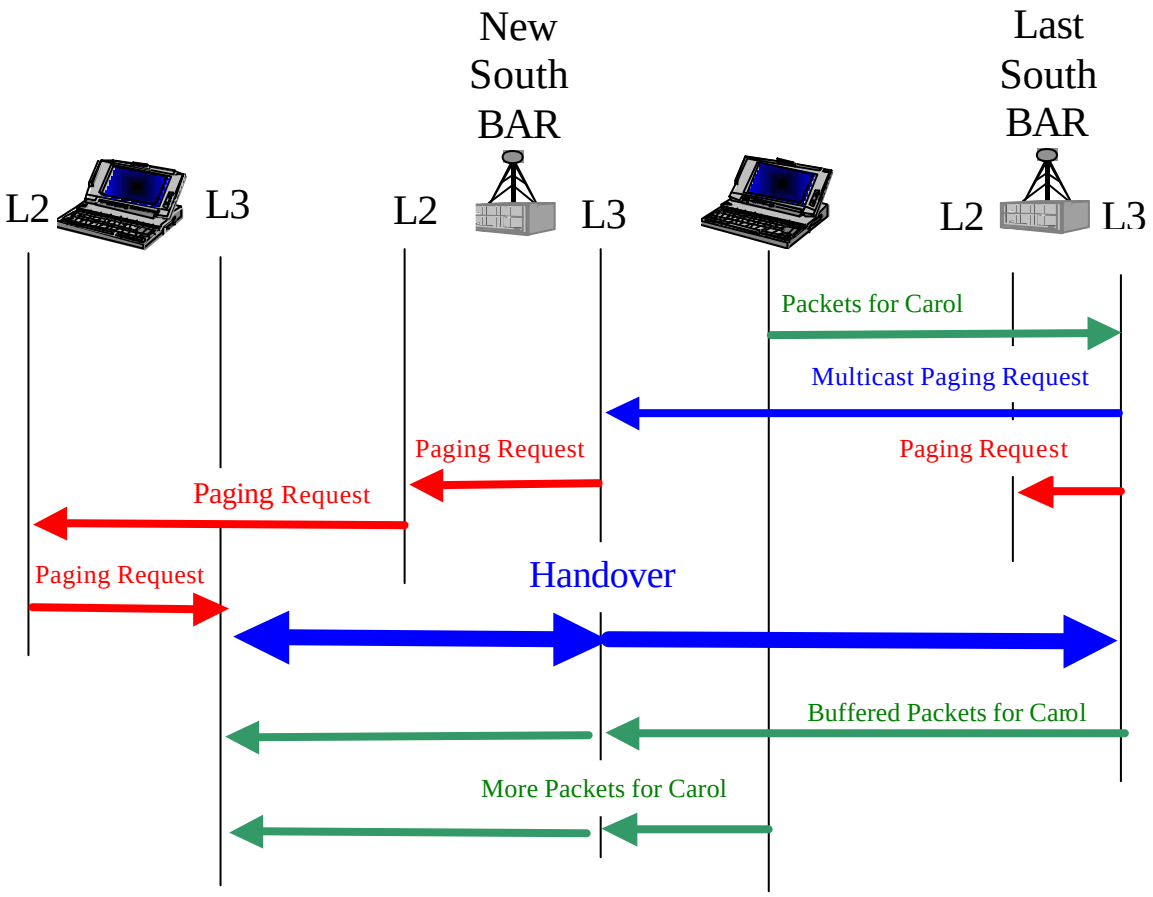


Figure 6-6: Paging Message Flow (Paging Request)

6.7 Conclusions

The 'Core' part of this report has summarised our conclusions on various network-layer issues: the BRAIN access network architecture, mobility management, quality of service, and the interfaces to higher and lower layers. In the 'worked example' above we have shown that our solutions can be assembled with standard Internet protocols to deliver the services required by a 'typical' user, Carol. The overall effect is a solution that supports mobile, wireless access to IP-based services in a way that complements 2nd and 3rd generation mobile systems and also keeps the benefits of 'traditional' Internet access. The follow-on project will trial some of the proposals presented here, with the results used to seed further simulations, as well as exploring other IP-based network developments.

7 References

[1.1] IST-1999-100050 project BRAIN, Technical Annex 1, “Description of work”, October 1999.

[1.2] IST-1999-100050 project BRAIN, Deliverable D1.2, “Concepts for Service Adaptation, Scalability and QoS Handling on mobility enabled IP networks”, March 2001.

[1.3] IST-1999-100050 project BRAIN, Deliverable D3.2, “Link and System Level Simulations and proposals of system optimisations for the standardisation process and a BRAIN follow up project”, March 2001.

[1.4] IST-2000-28584 project MIND.

[1.5] IST-1999-100050 project BRAIN, Deliverable D1.1, “Scenarios for mobile IP services and resulting requirements in different wireless networks”, August 2000.

[2.1] B. Carpenter, "Architectural Principles of the Internet," RFC 1958, June 1996.

[2.2] J.H. Saltzer, D.P. Reed, D.D. Clark, "End-To-End Arguments in System Design," ACM TOCS, Vol 2, Number 4, November 1984, pp 277-288.

[2.3] David S. Isenberg, “The Dawn of the Stupid Network”, ACM Networker 2.1, February/March 1998, pp. 24-31.

[2.4] IETF draft “Mobile IP Authentication, Authorization, and Accounting Requirements”, draft-ietf-mobileip-aaa-reqs-03.txt”

[3.1] Eardley P., Mihailovic A., Suihko T. “A Framework for the Evaluation of IP Mobility Protocols”, PIMRC ’00, London, September 2000.

[3.2] Keszei C., Manner J., Turányi Z., Valko A., “ Mobility Management and QoS in BRAIN Access Networks”, 1st International BRAIN Workshop, London, November 2000.

[3.3] IETF Code of Conduct <draft-ietf-poisson-code-02.txt>, “Scaling is the ultimate problem … many ideas quite workable in the small fail this crucial test”, this applies to any IETF protocol.

[3.4] Sarikaya B. Haverinen H., Malinen J.T., Magret V., “Mobile IPv6 Regional Paging”, <draft-sarikaya-mobileip-hmipv6rp-00.txt>, Nov. 2000.

[3.5] R. Ramjee, T. La Porta, and L. Li, "Paging support for IP mobility using HAWAII", Internet Draft (work in progress), draft-ietf-mobileip-paging-hawaii-00.txt, June ’99.

[3.6] Castelluccia C., “Extending Mobile IP with adaptive individual paging: a performance analysis”.

[3.7] Zhang X., Castellanos J., Campbell A., Sawada K., Barry M., “P-MIP: Minimal Paging Extensions for Mobile IP”, draft-zhang-pmip-00.txt>, July 2000.

A1 BRAIN WP2 ANNEX

This section presents the annex supporting the core report presented already. It's a list of project documents and published papers that led to the conclusions that were presented in the D2.2 core report. This section is organised relatively to the core report and contains:

- Architecture Annex where all the architectural documents and papers are listed.

- Mobility Management Annex with the documents relating to mobility management

- Quality of Service Annex where the material related to QoS is presented.

- Enhanced Socket Interface Annex with the specification of the ESI.

- IP2W Interface Annex with the specifications of the IP2W interface.

- Simulations Annex where the work relating to simulations is presented.

A2 Architecture Annex

A2.1 Modular IP Architectures For Wireless Mobile Access

This sections describes architectural paper that was writeen and presented in both BRAIN workshops in London and Yokosuka Research Park. This paper was co-authored by Phil Eardley and Robert Hancock

A2.1.1 Abstract

This paper describes the architecture of the BRAIN network layer. Firstly, the overall design approach is discussed, with comparison to the current paradigms of GSM/UMTS and the Internet. Next, the top-level architecture is presented, first looking at the way the access network fits into the end-to-end communication path, and then looking at current work on the internal structure of the network layer, with attention to the specific problems of link layer integration, mobility, and quality of service. We conclude with two examples of how functionality to support specific requirements is being designed within the context of this architecture.

A2.1.2 Introduction

A2.1.2.1 The Network Layer in the Context of the BRAIN Project

As has already been described in other papers in this workshop, the overall BRAIN project is a wide ranging research activity to develop an IP-based mobile wireless network complementary to current 2nd and 3rd generation systems. The initial focus is customer premises applications evolving from WLAN systems; however, it extends naturally to public metropolitan networks as the demand for broadband multimedia increases, and thus is a first step beyond 3G networks. The project encompasses user applications, through middleware, all the way to the air interface; the focus of this paper is the network layer architecture which supports and unifies the entire system. The key problems here are seen as the interactions between mobility and quality of service, the adaptation of applications and protocols to a wide variety of air interfaces with varying QoS, and the unification of a disparate set of Internet protocols into a coherent mobile network.

The requirements for the access network within BRAIN can be stated very simply as follows:

The basic goal of the BRAIN Access Network is to make mobile wireless Internet access look like 'normal' access through wired infrastructure.

The remainder of this paper explains what this really means in practice and how the problem can be structured to make it more achievable. It describes the process by which the BRAIN network layer architecture was developed, and how this top level architecture is now being refined into a concrete overall system design for future mobile wireless networks in the remainder of the project.

A2.1.2.2 Scope of the Network Layer

The BRAIN network layer encompasses both the terminal and the infrastructure of the access network. The scope of the BRAIN network layer is shown in Figure A2-1. In the terminal, it consists of an Internet protocol stack with backwards-compatible optimisations for mobile applications, and a lower convergence layer interface towards the selected radio technology. In the access network, it provides support for local mobility which is optimised for transport of IP application data, and makes the assumption of a direct interconnection with fixed IP backbone networks with a standard routed interface.

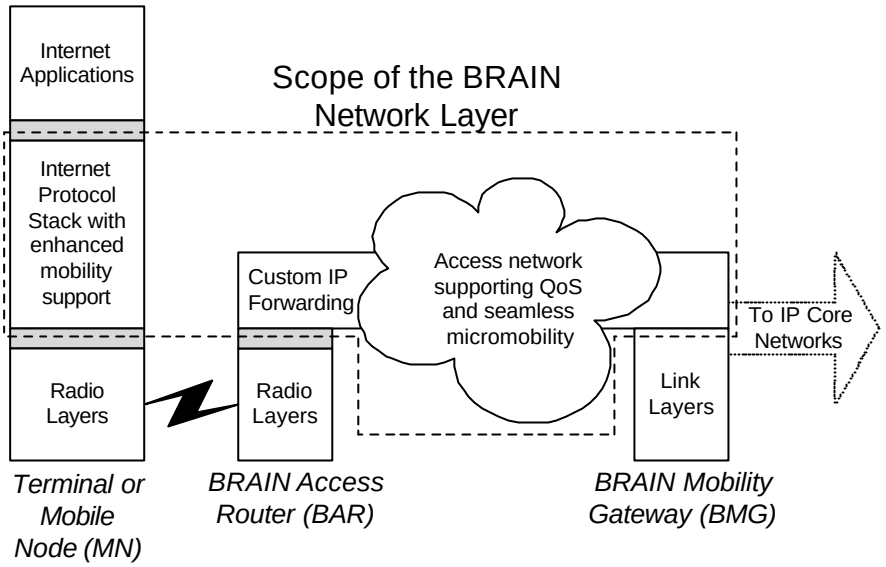


Figure A2-1: The Network Layer in the Context of BRAIN

A2.1.2.3 Structure of this Paper

It is important here to realise that a network architecture is more than just a 'bag of protocols': significant design choices have been made even at this stage, before consideration of the individual protocol components inside the network layer. Therefore, in the first section of the paper we will try to explain the rationale behind these choices and their implications. One particular choice is that the access network should be IP-based, and this requires both explanation (what does "IP-based" actually mean?) and justification.

The second part of the paper describes the key elements of the architecture, in particular interfaces to other components of a complete network (and interactions with other research activities, within and beyond BRAIN itself). This can be considered as the completion of the first stage of system design: the role of the access network and network layer have been defined, and the search can then begin for the major components required to build it.

The third part describes current activities that will refine this architecture into a complete and implementable system design. We present the major directions along which the design work is proceeding, and explain some of the interactions which have to be considered between the various components. As an example of this, we will consider in more detail the way in which the problems of handover and radio resource control – clearly the major distinguishing characteristics of mobile networks – are handled within the network system design.

A2.1.3 Motivation for the BRAIN Access Network

A2.1.3.1 The Drive Towards IP-Based Networking

The main motivations for the investigation of an mobile access network based on IP technology fall into three categories: those of interest to accountants, those of interest to engineers, and those of interest to end users.

Successive developments in mobile networking have always seen the network designers select the fixed network standard of the day as their choice for the transport infrastructure. GSM used the 64kbit circuits of ISDN; by the time its packet-based next step had been standardised, Frame Relay was the packet transmission mode of choice. In the meantime, it was clear that 3G networks would be intrinsically multiservice, and so the natural choice for terrestrial infrastructure was the broadband multiservice descendant of ISDN, namely ATM. All these choices have led to successful, high performance mobile networks – so, is there any pressing need for yet another change of direction towards IP?

The key point here is that, certainly in terms of traffic load, mobile networking is and will remain for some time to come a minority user of overall data communications networks, especially when future developments in residential networking are considered (e.g. cable networks, ADSL and so on). Therefore, it has always been assumed that, *for reasons of economy*, mobile network infrastructure should be based

on the prevalent fixed networking standard. Considering that even video entertainment is likely to be delivered over IP in the future, there is no doubt that IP is the correct future proof choice. There are two major aspects to this:

?? IP is or will be ubiquitous. In the near future, it can be expected that it will be possible to obtain broadband IP connectivity at any geographical location with relatively minimal cost. This is particularly significant for providers of radio access networks based on recent high bandwidth (but short range) air interfaces, for whom the cost of installing dedicated transport infrastructure would be prohibitive.

?? There are economies of scale, both in installation and operation. A single cabling and routing/switching network supports all mobile and fixed customers, public and private. Alternatively, the drive towards virtual private networking in the corporate market using techniques such as MPLS will lead to an open market whereby IP transport can be purchased and traded as a commodity, which will allow for low cost of entry and rapid deployment options for new operators.

So, these are the economic reasons. Although these do not impact directly on the type of mobile service that can be offered, there are also *sound engineering reasons* for the choice of IP as a universal network layer. There is a growing consensus in the networking community that the philosophy embodied in the IP protocol suite has benefits over more traditional (connection oriented, cell or frame switching) networks such as ISDN or ATM. The main aspects of this philosophy (and their corresponding advantages) can be summarised as:

?? Keep the network simple, and push complexity into end systems. This makes the network cheap to install and administer; in particular, the move away from connection based approaches minimises the requirement for very high availability within individual elements of the network infrastructure (one of the most onerous requirements for switching equipment).

?? Make the network modular, with open interfaces placed along natural functional boundaries. This makes the network functionality simple to evolve, since one part of the network can be upgraded independently of other parts. This allows new ideas and new technology to be exploited rapidly, without waiting for other parts of the network to catch up.

We will consider these ideas in more detail below, when use them as a starting point for deriving design principles for our own access network. However, again we note that these advantages relate to making life easier for access infrastructure developers, without necessarily changing the quality of the solution that can actually be developed. The final set of motivations relates to advantages which are visible directly to end users (or at least, affect the equipment which they own).

The assumption here is that in the future (and the not too distant future), all end user applications will actually be natively IP-based – that is, they will be written by people who take for granted the ability to send and receive IP packets. This can be seen in the current push from WAP and SMS messaging towards 'proper' Web and email access, and even voice traffic, the cornerstone of current mobile standards, is likely to be carried over IP eventually. The movement towards 'pure' IP based access infrastructure for mobile networks means that these applications available on fixed networks will inherently be available on mobile networks, *and furthermore that they will behave consistently there*, without their characteristics being submerged or modified by layers of mobile specific protocols. The same simplification will apply to user terminals: a single IP stack (with all that that implies for simplicity of security management, address assignment and other configuration) will be all that is necessary.

So, in the following, we will take as given the necessity for developing a mobile wireless access infrastructure which is based fully on IP technology, and we will also attempt to explain more concretely what 'IP-based' actually means in this context.

A2.1.3.2 The Need for a Consistent Network Architecture

It is already possible to build an IP-based access network. Protocols already exist for mobility support within IP, and also for quality of service control and configuration, and also secure communications. Why is anything else required than simply to deploy them all at the same time? Why is an architecture needed at all?

The first problem in fact is not a shortage of solutions, but an excess of them. There are many, many IP-based solutions for mobility support, running from simple roaming, through traditional Mobile IP, all the way to specialised schemes for local micro mobility support [A2.1]. These solutions are in no way equivalent, and so they cannot be considered as 'choices': once one has been chosen, others will likely require additional effort to support (and may even be excluded). Unless we want to accept a world of

incompatible IP radio networks, these multifarious solutions must be adapted to fit into a common framework.

The issue here is not limited to the fact that there are different solutions to the same problem. More, it is the case that there is no common agreement on what problem these solutions are trying to address. For example, in the area of security:

?? Some solutions are constructed as extensions to Mobile IP, which means that they can leverage the Mobile IP security capabilities – but they also require them to be present.

?? Some solutions are instead intended to be complementary to traditional remote access and so would use those existing AAA procedures (and also completely different assumptions about address management from the Mobile IP case).

?? Yet other solutions ignore the security issue altogether, and introduce subtle vulnerabilities which have to be analysed and 'fixed' by external means.

The problem of interactions is not limited to security. Indeed, a more serious interaction is with the area of quality of service. It is not generally appreciated that 'pure' IP mobility as currently described is in fact a step backwards in functionality from current 2G and 3G systems – whereas in those networks, a handover is always taken to mean moving a radio bearer from one place to another including all its performance characteristics, IP mobility simply considers the pure rerouting problem, with QoS handled by some completely different set of protocols. And yet, the interactions are very close, since (for example) the most appropriate style of handover may depend on the QoS requirements of the traffic being carried, and the mechanism for supporting QoS will require knowledge of the updated path.

As well as the issue that individual parts of a solution which have been designed and considered in isolation will not fit together, we must also consider that their appropriate scope of application may be very different, in an unhelpful way. For example, one solution may impact only on the wired infrastructure at the very edge of the network, while another may require special capabilities in the terminal as well. Both may require additional support from other protocols in other parts (e.g. the core) of the network, which a third approach may be a complete solution, but which has to be implemented completely – or not at all. This is particularly the case for IP mobility when compared with the architecture of 2G and 3G networks: an IP mobility protocol might be an optimal replacement simply for radio access network mobility or GPRS core network mobility as well, or it may be most appropriate for inter-operator mobility instead.

Once we have accepted the need for an architectural framework within which to address the access network problem, there are further advantages to having a design which is relatively prescriptive in terms of functional interfaces. The advantage is of efficiency and performance. Where parts of the solution are designed in isolation, they cannot make assumptions about supporting capabilities that will be present (albeit developed independently). This is a particularly severe issue in the case of mobile wireless networks, where (for example) the characteristics of the link and physical layers can have a very significant impact on the overall user-perceived performance. Adoption of a concrete architecture, with well defined inter-layer interfaces, means that system functions can be implemented wherever it is easiest to do so, and yet made available to the other components in a well defined way. It should be noted that this is a departure from the 'traditional' approach which has been used in the design of many aspects of the Internet, where the emphasis has been on interoperability at the individual protocol level.

A2.1.3.3 Design Goals for the Access Network

This section presents the design principles that will be used to guide the selection of the basic architecture of the BRAIN network layer. This includes both the access network infrastructure and the network layer in the terminals themselves. The list is not definitive or exhaustive, nor does it replace the formal requirements on the access network; instead, the intention is to describe the motivation for the decisions that have been made where several alternatives are available.

A2.1.3.3.1 The End-to-End Principle and Transparency

The end-to-end argument is one of the architectural principles of the Internet [A2.2] [A2.3]. The basic argument is that, as a first principle, certain required end-to-end functions, like end-to-end reliability and security, can only be correctly performed by the end systems themselves. In order to a support this, the network should offer only some kind of minimal service to the end systems. In addition, providing specific functions within the network often also makes that network hard to evolve towards support for new services. Therefore, most functions are best implemented in the end systems themselves and network implementers should concentrate on providing a basic service, with optimisations (if any) for

performance. (Following these guidelines means that the end-to-end principle in designing the Internet protocols is retained as the prevailing approach.) The end-to-end principle is sometimes reduced to the concept of the 'stupid network' [A2.4]. In the mobile environment, the term is unfortunate since it is very hard for a high performance mobile access network to be truly stupid. Nevertheless, the underlying concept of minimal network functionality still applies (that is, the network should still look stupid to outsiders).

Thus, in the context of a mobile access network, this principle can be refined more concretely as follows:

?? Be independent of specific transport layers and applications. Provide only a connectionless packet service, which offers (with varying degrees of performance) to get packets from A to B. In the scope of this project, we restrict ourselves to the Internet protocols, so we only consider IP packets.

?? Be as independent as possible of what type of packets are being transported, and assume simply that packets are forwarded according to their IP header. In particular, try not to depend on specific properties of IPv4 and IPv6, and don't assume that any mobility encapsulation is used above them.

?? Minimise the number of special functions that are provided in the access network. The main role of the mobile access network is to look like a (genuinely stupid) fixed access network, that is, hiding mobility. Mobility support, especially for fast handover, can best be provided by the network and not just the end system, but this should (and can) be done in a way which does not alter the transparency of the network. New externally visible functions should be limited to mobile-specific ones (such as location dependent service support).

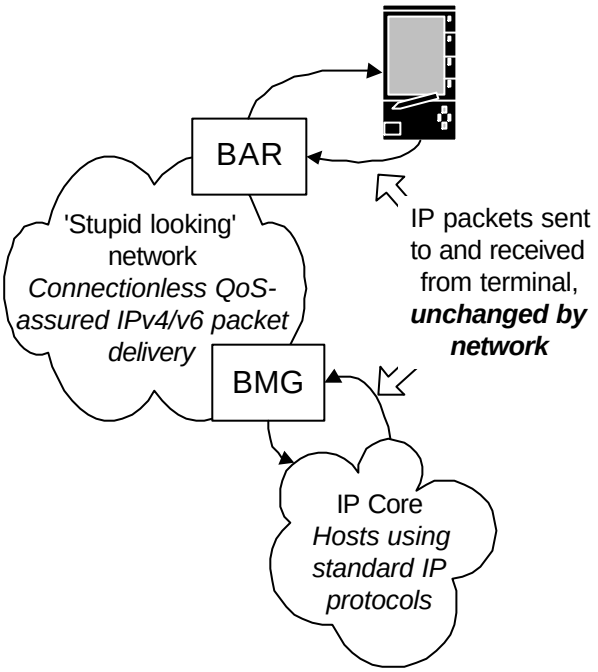


Figure A2-2: The Transparent Network

In other words, the access network is a machine for delivering IP packets – *and doing nothing else*. This mindset is illustrated in Figure A2-2.

In this context, it is possible to take a 'two-level' view of the 'end system' and the heart of the network:

1) At its most basic, the 'end system' is the terminal, and it sees the whole of the rest of the network simply as providing packet transport. The emphasis here is on making this entire network as transparent as possible.

2) As a refinement, there is a similar distinction between the 'fairly smart' access network, and the very simple core network. The access network provides some specific extra functionality mainly associated with mobility, while the core is restricted to pure packet transport capabilities.

This principle does not provide a clear answer to the question of where within the network hierarchy functions such as authentication, and particularly authorisation and accounting are placed, and this is discussed in more detail below. However, it is clear that the access network should not attempt to provide

strong security for user-user data since this can only ever be assured end-to-end. Access network security for this type of data should be limited to providing 'wired equivalent' security over the air interface.

A2.1.3.3.2 Obey the Layer Model

The layer model is a fundamental principle to be used in clear protocol stack design. Specifically in this context, this implies that access network should limit its functionality to providing IP packet forwarding, independent of upper layer applications.

In mobile networks, it is common to end up with different layers of the protocol stacks being tightly integrated together in the interests of efficiency. In the case of BRAIN, we should aim to structure things in a more modular fashion with clear inter-layer interfaces. Specifically:

?? The network layer within the access network should have a generic interface towards the link layer, such that new (and old) link layers can be exploited without wholesale network infrastructure redesign.

?? Where particular applications require optimised support, this should be invoked and made available in a generic way – typically via some sort of QoS aware service interface. All link layer specific features should be hidden as much as possible from the upper layers.

Note that some useful information or 'hints' about the link layer internals can, and even should, be made available for the upper layers to allow efficient operation above the link (e.g. amount of buffer space at link layer and indications of upcoming buffer exhaustion so that upper layers may react, or preferably, take proactive actions as seen necessary). However, this should be done in a way which does not change the upper layer protocol semantics, so the impact is limited to the implementation within a single network element.

A2.1.3.3.3 Maximise Future Flexibility and Evolvability

It is desirable in any network to minimise any barriers to technology evolution. This applies equally to upper layer services, link layer technologies, and indeed components of the access network that lie between these. A related principle is that it should be easy to deploy a new system incrementally, for example, an initial system with limited performance followed later by performance extensions.

This is a particularly strong requirement in the public mobile environment, where a system upgrade which changes interfaces may involve hundreds of different organisations each with their own infrastructure and hundreds of millions of terminals. Some specific implications of this are as follows:

?? Components within the access network should be modular, so that different parts can be evolved and upgraded independently. Interface between the components should be clear and well defined.

?? For example, the logical interface between the terminal and network which signals handover events should not enforce the use of a particular micro mobility protocol, but should allow network providers to choose appropriate solutions depending on their business model and deployment environment. This is illustrated in Figure A2-3.

In general, this approach will allow for rapid development of initial solutions, while not ruling out performance enhancements in the future. These can be provided transparently to the end users.

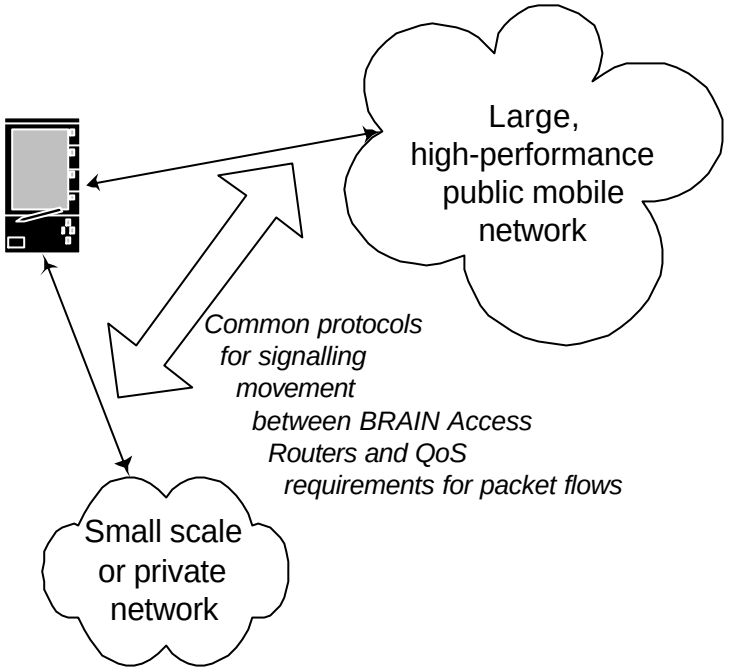


Figure A2-3: Application of a Common Air Interface Signalling Protocol

A2.1.3.3.4 Minimise Requirements on the Terminal

It is clear that minimising terminal requirements sometimes pulls in the opposite direction from the end-to-end principle. On the other hand, it is clearly very desirable in the constrained mobile environment to minimise the demands made on the terminal. So we can rephrase this as:

?? Minimise resource demands on the terminal.

This principle leads for example to the requirements that the access network should support optimisations for idle modes, and should try not to impose a heavy signalling load to support mobility.

A2.1.3.3.5 Don't Re-Invent the Wheel

Where protocols already exist for a particular problem, these should be re-used unchanged if possible or extended otherwise. This particularly applies to protocols which extend into the fixed network or application layers in the terminal.

Where protocols or functions (e.g. IP routing and forwarding) are required within the access network, standard solutions should be re-used if possible to maximise the scope for infrastructure sharing with existing networks. If this is not possible (e.g. for performance reasons), any new protocol should try to follow the same approach as used in the Internet today – i.e. to aim for a connectionless, stateless, resilient and highly distributed model.

A2.1.3.3.6 Exploit Standard Functionality in the IP Core Network

This means that we don't require any special mobility (i.e. handover-related) support in the core network, and are prepared to work with whatever QoS solution the 'local' core network provides. Additionally, we attempt to maximise the re-use of standard functionality for which core network protocols are already available – especially in areas such as application call control, end-to-end security, authentication, and so on.

A2.1.3.3.7 Keep it Simple

The key point here is not to complicate the system by attempting to reproduce optimal solutions for every feature that exists in current mobile networks, especially where to do so conflicts with the other requirements above.

A2.1.4 Overall Structure and Requirements for the Access Network

In this section, we consider at a very high level the way in which the functionality needed in the access network fits into the larger mobile networking picture. There are four major aspects to this:

1) The fundamental quantity which is used in the network layer to deliver packets is the network address – in our case, the IP address. What is the scope of these addresses and how are they controlled and assigned?
2) Just how big is an access network, and where are its boundaries?
3) How does the access network interact with other (fixed) networks for security purposes, i.e. to control access by users and generate accounting information?
4) How does the network layer relate to the other layers (upper and lower) which are needed to support applications over wireless physical layers?

Note that we don't consider *internal* issues like micro mobility support and access network internal QoS handling at this stage: the idea is that provided the above questions about external interfaces have been pinned down, mobility and quality of service can then be considered largely according to mechanisms specific to the BRAIN access network. Since both correct routing and quality of service are only of value if they work end-to-end, this is of course a simplification which might lead to a sub-optimal solution compared to an integrated end-to-end approach; however, it is consistent with our guiding principle of wanting to leave the core IP network unchanged as far as possible.

A2.1.4.1 Addressing

As stated at the outset of this paper, the basic goal of any given BRAIN Access Network (BAN) is to make mobile wireless Internet access look like 'normal' access through wired infrastructure. Thus, a BAN must allow a terminal to get an IP address to use in communicating with correspondent hosts in other networks; the BAN routes packets to and from this address in a way which externally looks the same as any other IP network.

The mechanism of address assignment has not been fixed, although solutions such as DHCP are one typical option; in any case, this is a function of the link convergence layer, which is discussed below. One assumption for BRAIN is that the address is unique to the terminal, rather than shared (e.g. as would be the case for 'foreign agent care-of addresses' of Mobile-IPv4). This is a consequence of the requirement for a clean, unified solution that applies to both Mobile IPv4 and IPv6 (and indeed many other higher layer protocols), recognising that shared addresses are simply one mechanism for IPv4 address space conservation, which is often ruled out because of security and other considerations.

This approach, of routing to the mobile based purely on an assigned, local IP address, is shown in Figure A2-4. It can be seen that the entities within the BRAIN access network operate as pure IP routers (at least so far as packet forwarding is concerned), with no special treatment for encapsulation or decapsulation of 'home addresses' of the mobile node. This does not mean that Mobile IP is excluded, just that it is optional within the terminal (in which case no Home Agent is required either), provided collocated care-of addresses are used. Indeed, the use or otherwise of Mobile IP would not normally be visible to the access network, unless it was to analyse inner protocol headers to discover the additional layers of encapsulation.

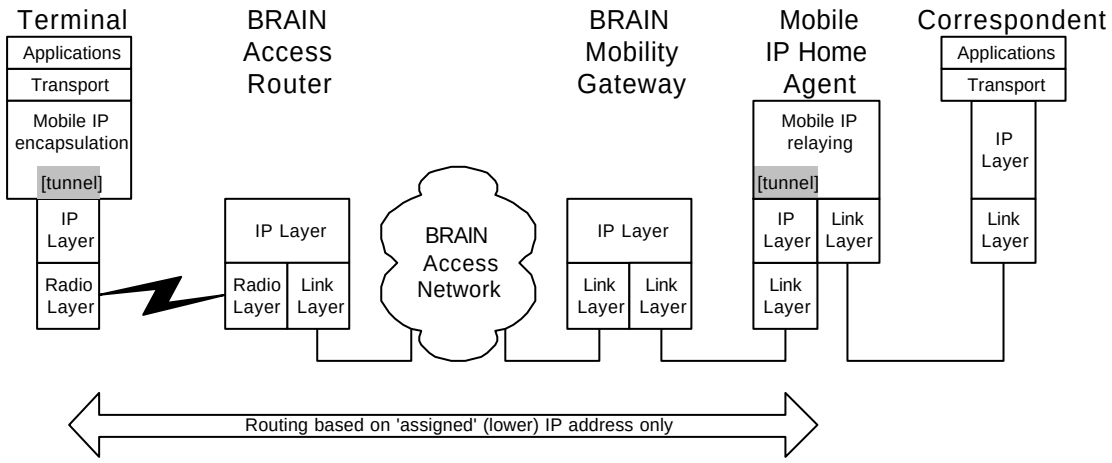


Figure A2-4: End-to-End Address Assignments

A2.1.4.2 Scaling

Once an address has been assigned, the fundamental role of the BAN is to support seamless mobility of the terminal as it moves between access routers. In consequence, the allocated address must remain valid throughout the entire BAN, so there is a direct relationship between access network scalability and address allocation. There are essentially two options:

?? If seamless mobility within a single geographically limited area only is required, a BAN is allowed to interconnect with the core network at a single point, corresponding to a single BMG.

?? If seamless mobility over a very wide area is required, the performance of the Internet prevents us relying on BAN-BAN handovers to support this. Therefore, the combination of wide area support and seamless terminal mobility forces the use of multiple interconnects with the core.

This is one example of using the option for different protocols within the BAN depending on service provider requirements, since achieving very high scalability for a terminal mobility and QoS protocols is a hard problem and not relevant to (for example) a campus network operator. In either case, it is assumed that a BAN is under single administrative control, and seamless handovers between administrations are not catered for. The combination of these scenarios is shown in Figure .

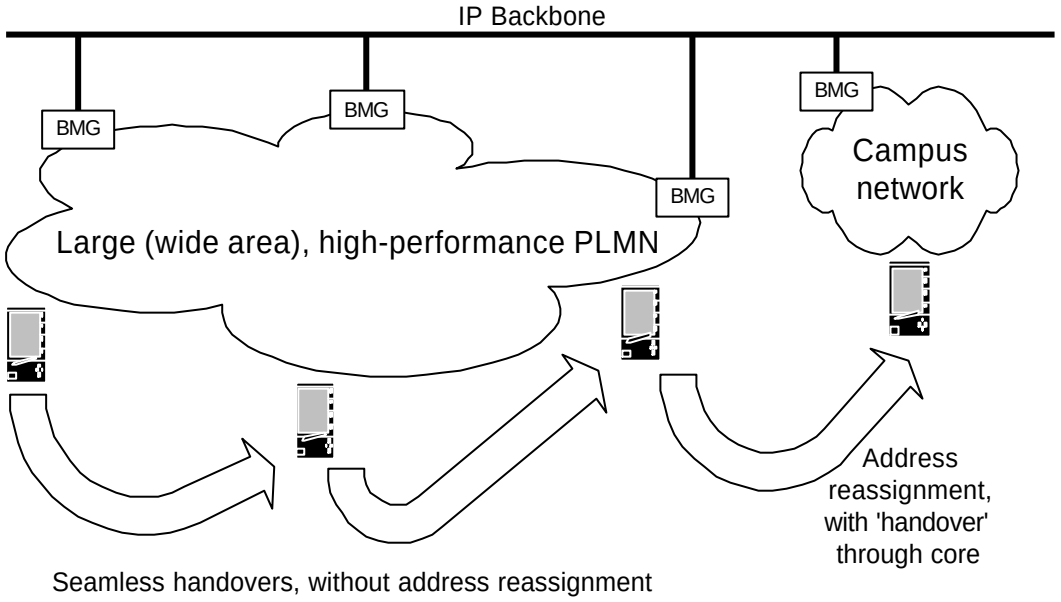


Figure A2-5: Access Network Scalability

A2.1.4.3 Security

In line with our design goal of keeping the access 'minimally functional', the main security issue which relates to the BRAIN access network is that of Authentication, Authorisation and Accounting (AAA) – the extent to which the BAN ensures security of user data is assumed to be limited to the level of comfort offered by today's mobile networks and wireless LANs (e.g. using relatively simple encryption over the air interface).

There is already a large and sophisticated set of standards to support AAA within the Internet, much of which has grown up around the need to support traditional dial-up access [A2.5]. These standards already support such advanced concepts as inter-ISP roaming, and are being further extended to such capabilities as hot billing and pre-pay. It is clear that most of these functions should be directly carried over into the BRAIN environment – indeed, given the requirement to appear as similar to fixed access as possible, it is almost mandatory that BRAIN should attempt to maximise re-use of the corresponding protocols and standards.

In the context of the BRAIN access network, there are therefore two major aspects of AAA which need to be considered:

?? A roaming user needs to be able to present credentials to the network which allow the user to be authenticated, and which allow the network to determine the resources to which the user is entitled. This is almost exactly the same problem as in the current fixed network case.

?? Additional requirements coming from mobility (and in fact, the desire to support billable QoS guarantees) are that the network should be able to validate messages requesting a handover (respectively, new session characteristics) actually do come from the user in question. And importantly, it should be possible to exchange these messages and carry out the validation very close to real time.

In keeping with the concept of re-using the current fixed network protocols, the BRAIN access network also carries over much of the current fixed network logical security architecture, in terms of AAA servers and (optionally) AAA brokers mediating between them in the case of roaming. The basic picture is shown in Figure A2-6, which also shows the trust relationships that have to be statically configured (or dynamically established) between the various entities. Note that the 'local' AAA server (AAAL) is shown as being part of the access network; however, it is assumed that this should be very similar to the corresponding device in a fixed access network, especially given that the protocols which are used between AAA servers and access routers (typically RADIUS [A2.6] or DIAMETER [A2.7]) are designed to be intrinsically extensible.

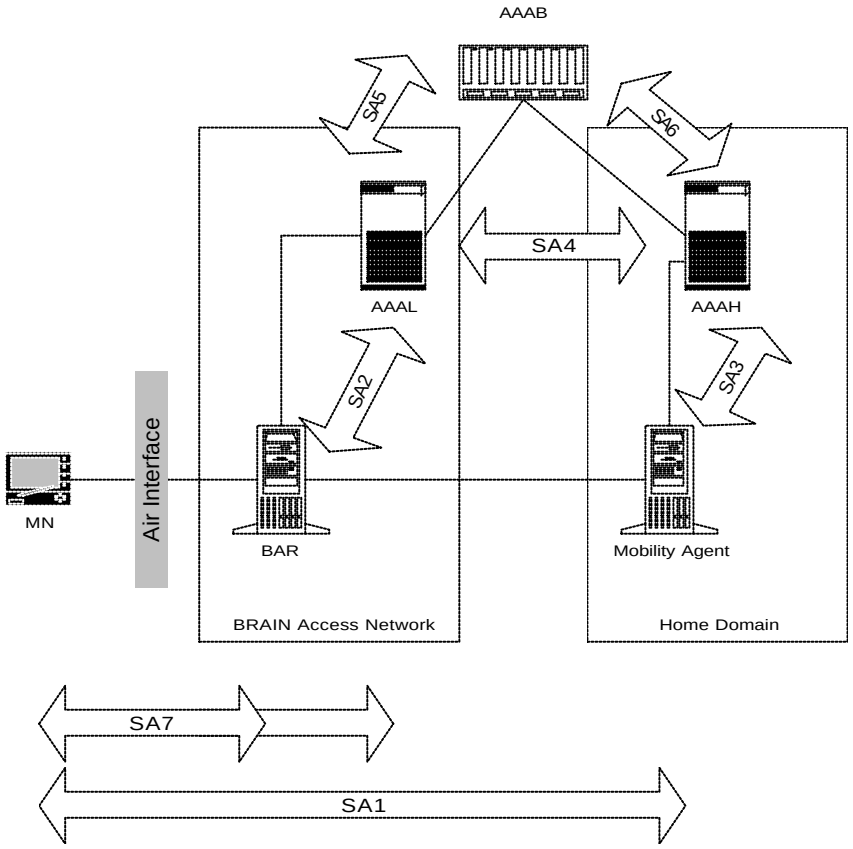


Figure A2-6: Security Infrastructure Supporting Access Network Operation

The trust relationships in the figure are shown here as security associations ('SA'), although it is not implied that IPSec is always used; in particular, existing AAA protocols generally have their own built in mechanisms for mutual authentication (as does Mobile IPv4, although Mobile IPv6 does use the services of IPSec, and this is a natural option to consider for other trust relationships where support for these would otherwise have to be designed from scratch). The trust relationships in detail are as follows:

?? SA1 is the trust relationship between an MN using some macro mobility protocol and the mobility agent in its home domain – it is therefore only present when this protocol is being used, and its existence is generally invisible to the BAN (because of transparency). Although it is usually statically configured, some mobility protocols like Mobile IPv6 include a mechanism for a mobile to dynamically learn the address of its home agent. In this situation, SA1 is no longer static but must be dynamic.

?? SA2 (and SA3) are used to secure network-internal communications between different entities within the same administrative domain. They can be considered as part of the management plane of the networks in question, and will usually be a matter of local choice.

?? SA4 is the trust relationship required to support roaming, and can either be statically configured between AAAL and AAAH, or established dynamically using the services of a AAA broker AAAB. In this case, SA5 and SA6 are required. Note that although these trust relationships cross the access network boundary, it is assumed that absolutely standard fixed network protocols can be used for these cases.

?? The most important trust relationship for the access network is SA7, and this is also the one which shows most differences from the fixed network case. The figure shows it running between the MN and access router, or possibly to the AAAL; in fact, the most that can be said at this stage is that it runs between the MN and some entity in the BAN. Issue that need to be considered about SA7 are that

  ?? it must offer good security (it is the foundation for allowing access to the network and ultimately for charging the user);

  ?? it must be possible to secure traffic between the MN and BAR, such as traffic related to handover or resource requests, with minimal delays for decryption and validation within the network – which would tend towards terminating it at the BAR;

  ?? it must be possible to move the MN's point of attachment to the network rapidly without requiring an extensive renegotiation of SA7 – which would tend to terminating it at a more central location.

In any case, it can be seen that there is a well defined set of external interfaces and 'outward facing' security components which can be re-used from current fixed IP networks to support the requirements of the BRAIN access network. At the same time, it can be seen that while the BRAIN specific requirements are challenging, they can also be decoupled from the external interfaces and considered specifically in terms of the MN-BAR air interface protocols and access network internal structure. This is one of the benefits of having a clear BRAIN network layer architecture.

A2.1.4.4 Inter-Layer Interfaces

In an activity concerned only with interoperability, there is no need for inter-layer interfaces since these can be considered as implementation issues. However, abstract interfaces play a valuable role in partitioning the mobile networking problem, and clarifying the behaviour expected from or supported by particular network components. By extension, they provide a framework for research into the operation of particular functions (for example, header compression or TCP performance). In the Internet world, service interfaces have traditionally been minimalist; however, enriching the functionality of these interfaces is one mechanism for allowing network performance enhancements towards the level of traditional PLMNs while preserving layer separation.

The BRAIN network layer relies on two inter-layer interfaces for this purpose. The first lies above the basic network and transport protocols and provides the enhanced application support that is necessary in the mobile environment. Broadly, it allows for extended negotiation of QoS information between the application and lower layers, including renegotiation during active sessions. It exists only in BRAIN terminals The second is a specialised interface for matching the IP layer to wireless layers, hence the name 'IP_2W', and is common to terminals and access routers. The interfaces are shown together in Figure A2-7.

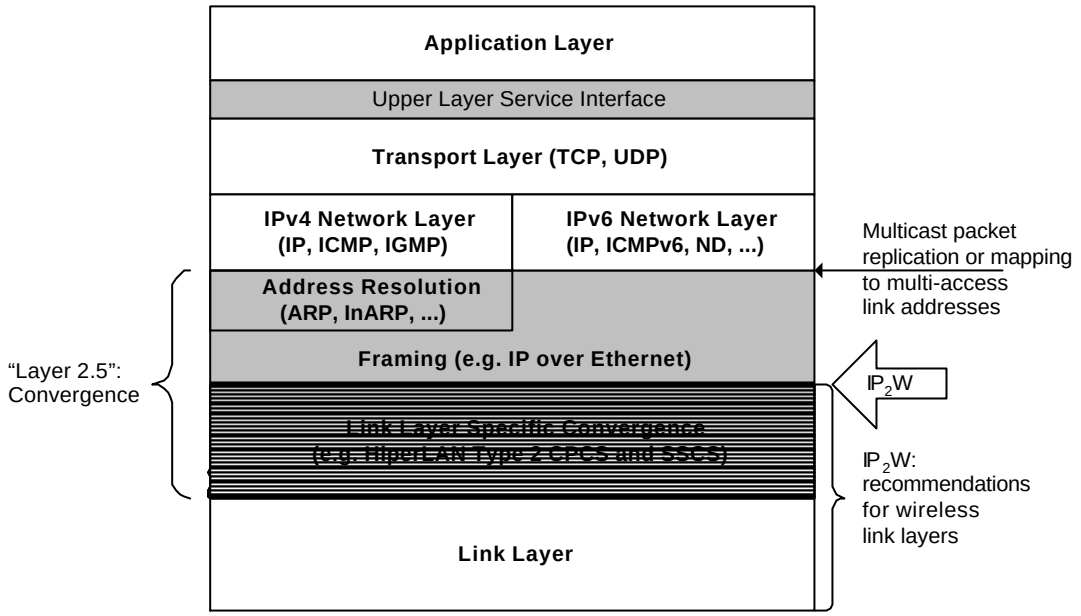


Figure A2-7: BRAIN Inter-Layer Interfaces

The combination of these two interfaces is the key to allowing the development of a re-usable, QoS capable "IP stack" which offers advanced facilities to applications, yet is efficiently integrated into link layer. The main issues at this interface are link set-up / release, layer 2 and 3 address assignment, link layer QoS negotiation and re-negotiation, and the interaction between this and buffer management (scheduling) in the network layer. In particular, IP_2W enables the use of layer 2 procedures which are much more efficient than equivalent IP protocols operating over a generic data interface. The performance of these operations and the level of control that upper layers have over them has a direct impact on the performance of handovers at the IP layer and on the QoS received by a mobile user.

In detail, the IP_2W interface is separated into a Data and Control part, each offering access to some functionality at the link layer. Several distinct functions have been identified under the interfaces; some are optional, and the link layer advertises which it supports through a configuration interface. The control interface is also used to control the operation of some of the user plane parts such as buffer sizing and error control characteristics. The model mandates no specific structure within a given link layer, and indeed, some functions may be inherent in a particular link type, while others may have to be added by a convergence layer. Where an option is not supported, the "IP stack" can fall back to a layer 3 protocol instead.

A2.1.4.5 Summary

It should be pointed out that, although we appear to have described a very abstract, high level approach, in fact the access network architecture presented embodies some very real and very significant design choices. These choices are by no means universal in all models of IP-mobility networking, and yet they are the result here of following a well established set of design goals and taking a clear view of the practical requirements that such a network has to follow. It is worth summarising the main issues here.

1) The access network needs some enhancements to the standard protocols of the fixed network. These are (only) to support mobility. The main role of the mobile access network is to look like a (genuinely stupid) fixed access network, that is, hiding mobility.

2) To achieve this a new network is required, with some degree of specialisation in two main areas:

   a) A way of distributing and updating information on the location of the MNs. The next paper [A2.8] discusses options in detail.

   b) When a MN hands over from one BAR to another, there must be a way of transferring state information associated with that MN. Section 5.1 discusses this briefly.

3) In particular, the access network is not based around a central assumption of ubiquitous Mobile IP (v4 or v6) – indeed, the use or otherwise of Mobile IP is almost orthogonal to the main concerns of the access network.

4) The scaling requirement for wide area seamless terminal mobility forces the acceptance of topologies which have multiple points of interconnection with the fixed network. Most of the current proposed micro mobility protocols are not capable of this mode of operation since they choose to limit themselves to the (much simpler) single attachment scenario.

5) A wireless/mobile optimised, feature rich layer 2 service interface is required for both terminals and access routers, which supports any air interface. There should be an aim for this service interface to be universal in mobile wireless networking. Such an interface enables a modular structure, so that new link layers can be exploited without wholesale network infrastructure redesign.

6) The scope of the network layer consideration should include the terminal. The "IP stack" functionality within the terminal can and should be enhanced in a backwards compatible way to make best use of wireless mobile networks.

In summary we believe that it is possible to define a useful, minimal functionality wireless mobile network which attaches to the IP core like any other access network. This access network should offer simple IP packet transport with QoS guarantees.

A2.1.5 An Outline Design for the Access Network

A2.1.5.1 Overview

This section presents an initial outline design for the internal structure of the access network, in terms of its primary components, and the way they fit together. It should be noted that this is still work in progress within the BRAIN project, and as analysis of the problem and current solutions proceeds, it can be expected that some of the details – maybe, very significant details – will change. However, it represents at least a self-consistent picture of a complete access network design, and can be used as a starting point for consideration of the possibilities for how an all-IP radio access network would really look.

The main components of the network layer problem are taken to be as follows:

?? Pure terminal mobility management – how to manage routing of IP packets around the network as terminals move.

?? Quality of service – how to ensure quality of service is provided to user sessions at the network level (and that these requirements are communicated to the lower layers). QoS includes the specific sub-problem of connection admission control – how to decide whether to admit a request.

?? IP_2W – the detailed design of the service interface between the network and link/physical layers. This implicitly extends a more general consideration of the protocols between the terminal and access network, including the question of how they are secured.

?? Application layer service interfaces – this covers the way in which applications running on the terminal convey their requirements (generally QoS related) to the network layer.

?? Radio resource management – the issue of how radio resource optimisation over the whole network (i.e. where the question is not restricted to single cell environments) interacts with the network layer.

?? Authentication, authorisation and accounting – how AAA functions support the above.

Note in fact that some design decisions have implicitly already been taken in identifying these as components of the problem, and as interactions between them are studied in more detail it is possible that some functional boundaries will shift. Nevertheless, they form a starting point for analysis. There is no assumption that the components of the solution will be the same.

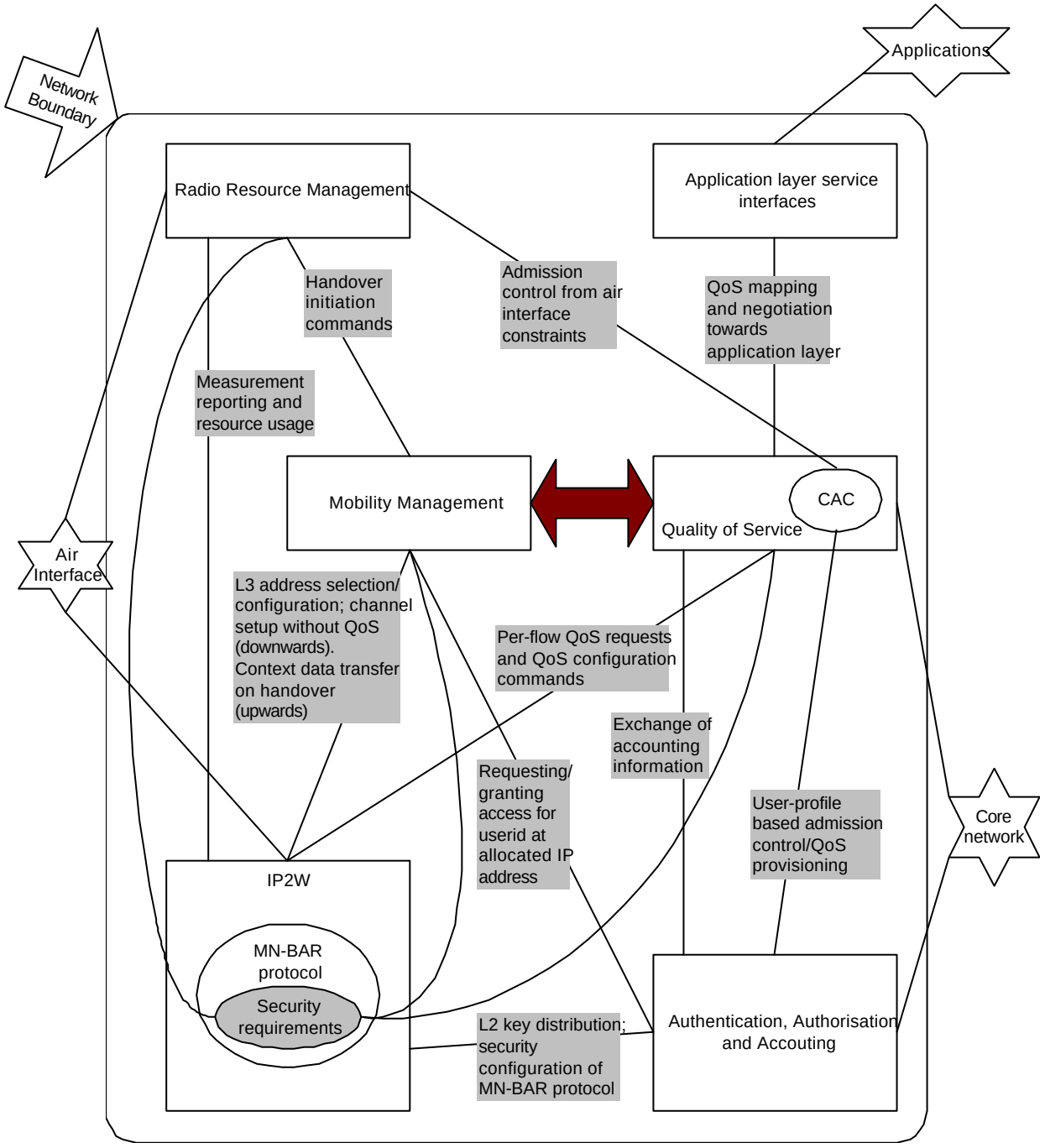


Figure A2-8: Interactions between Network Layer Component Problems

The main interactions between these various problems are shown in
Figure A2-8, which also shows the interactions with aspects 'external' to the access network – namely the lower layers, core networks, and applications. In some cases it is possible to pin down the interactions quite precisely, for example the fact that admission control can be expected to take inputs from radio resource management (on the current state of the air interface) and AAA. Other interactions are considerably more complex. Notable here is the interaction between mobility management and the quality of service problem. As has already been pointed out, it is only in IP networking that these are even considered separate problems, and analysis of how to keep them separate is still continuing.

More detailed papers and presentations on many of these problems are being presented separately at this workshop. Here, we give a simple overview of the scopes of the main component problems [6].

A2.1.5.2 Mobility Management

The problem to be solved under the heading of mobility management can be very simply stated:

1) To decide what IP address to allocate to the user, and interact with the control plane (basically, AAA procedures) to control users attaching to the network.
2) To maintain routing information within the access network to allow packets to and from that address to be routed correctly between the appropriate BAR and BMG (and thence to the core network). Similarly, to enable transfer of other state information around the network in support of handover.
3) To do all this with the appropriate performance requirements (reliability, idle mode support, scalability and so on).

There are several well defined interaction points (with AAA for security, with IP2W for security and L2/3 address mapping, and with radio resource management for handover initiation) and one very broad interaction point (with QoS). These are all discussed below.

A2.1.5.3 Quality of Service

The QoS problem is more difficult to lay down, because there are more aspects of it. The main ones are:

1) To decide what QoS service classes should be supported and how they should be specified and mapped at various levels of the MN and BAR. As has been stated above, deciding the level of sophistication in the QoS classes to be supported is the result of a trade-off between future proofing the solution against all possible application requirements and on the other hand minimising network complexity.
2) To establish a mechanism by which the QoS requirements of application layer flows are communicated to the access network infrastructure, such that the access network CAC entity can decide whether to assign the requested QoS guarantees.
3) To implement these guarantees within the network and notify the radio QoS requirements through the IP2W interface.
4) To do all this again through handovers. Note that the style of handover (low latency or loss free) maybe depend on the QoS class of the underlying data stream.
5) To do this all with appropriate performance requirements.

As for mobility management, there are several well defined interfaces (to IP_2W and radio resource management, possibly to AAA for user-based admission control), and the broad interface (to mobility management).

A2.1.5.4 IP_2W

The IP_2W task is the most important and complex of all the tasks. We can break it down as follows:

1) To define the primitives across the IP_2W interface.
2) Implicitly, to define the MN-BAR protocol at the IP_2W level (i.e. matching primitives each side of the air interface for e.g. address negotiation, security key setting and so on).
3) To define a mechanism whereby the state below the IP_2W in one BAR can be transferred to another BAR (this forms part of the inter-BAR handover protocol, which itself is really part of the MM task).

To define what should be implemented below the IP_2W for the HIPERLAN/2 case.

The IP_2W task interacts with everything, but mainly by taking inputs and defining syntax/semantics. The interactions with mobility management, QoS and AAA relate mainly to configuration of the IP_2W; IP_2W is also a transparent conduit for information to the radio resource management subsystem.

[6] Apart from the application layer service interfaces, which simply mediate between application layer requirements and the services provided by the "IP stack" within the terminal. In addition, the radio resource management aspect is considered in section A2.1.6.3, since this is mainly a more detailed matter of interface design.

A2.1.5.5 AAA

Security aspects fall into two almost independent parts.

1) The AAA subsystem, which does access control and is involved in constructing a secure association with the MN for authenticating BAN-related control plane transactions. The precise mechanism for doing this is still very open (as has been mentioned above, there are conflicting requirements for the best way to do it).

2) AAA support to IP_2W for configuration of air interface security.

As well as this, the AAA subsystem has an interaction with QoS whereby it can take part in the admission control decision (allowing decisions to be based for example on user identity) and also, in the future, accepting accounting information from the QoS control entities within the network.

A2.1.6 System Design for Key Features

A2.1.6.1 Introduction

This section presents a more detailed discussion of issues and requirements associated with handover and radio resource management. It should be noted that this is still work in progress within the BRAIN project. This represents the current analysis of the problem, but as work proceeds, it can be expected that some of the details – maybe, very significant details – will change.

A2.1.6.2 Handover types

A2.1.6.2.1 Introduction

The objective of a handover is to change the point of attachment, the carrier/code, the technology or the network, whilst maintaining a MN's session. In this paper we only consider handovers that affect the IP layer, i.e. where the BAR changes; we do not consider a "layer 2 handover",[7] since it is transparent to the IP layer.

There are a number of issues relevant to handover, some of which are 'compulsory' and some 'optional'. Many of these are discussed in more detail below, but first we briefly introduce them. Those issues involved in all handovers are:-

?? **Handover decision**
Section A2.1.6.2.2 discusses what roles we believe the MN and network should have in the decision making process that determines a handover is required at a particular instant.

?? **Connection changes**
New connections (radio and fixed) need to be set up and superfluous connections released. This will take place at several levels of the protocol stack (layer 1, 2 and 3).

?? **Re-routing**
Handover implies re-routing of connections through the fixed networks, potentially even outside the currently involved access network.

Amongst those issues that may be involved in some handovers are:

?? **Information gathering and reporting**
The BAR and/or MN may measure the radio quality (e.g. to detect if a new BAR is coming in range. Information may be gathered about radio resource usage in other nearby BARs.

?? **Address negotiation**
A mobile node may need to acquire new L2 and L3 addresses during handover.

?? **Diversity combining**
If "soft handover"[8] is supported, then connections are added to and released from combining/splitting points. In this case, adding a connection does not imply releasing another one.

?? **Packet loss alleviation**
In order to guard against packet loss, packets can be duplicated or buffered.

[7] We define a "layer 2 handover" as: a handover during which a MN stays connected to the same BAR but changes access points (or some other aspect of the radio channel).

[8] "Soft handover" is defined and discussed in Section 4.4.

?? **QoS revalidation/renegotiation**
The QoS of the radio channel in the new cell, and the new path through the network, may need to be set up. As part of this, QoS violations may need to be taken into account and a QoS renegotiation take place.

We now consider three questions relating to handover:

?? Who controls the handover?

?? What handover performance does the user require?

?? How can the network deliver this performance?

Our perspective is mainly on what the requirements are, rather than on implementation issues.

A2.1.6.2.2 Who controls the handover?

We believe that handover should be mobile controlled, i.e. the MN initiates and decides about a handover[9]. This is for a number of inter-linked reasons:

?? The MN is best placed to understand the application's needs, the current competing requirements of different user processes / OS activities, and the personal preferences of the user. This argument is increasingly important in a multi-service world, whereas historically the only application has been voice, when it is relatively easy for the network to guess what the MN would like.

?? Complex charging schemes, and in particular receiver charging, encourages MN control. For example, the opportunity to handover to a higher quality, but more expensive link, and the possibility to receive data "pushed" by some distant server, but where the MN will be charged for transmission over the air interface – only the MN can decide whether it is prepared to pay. By contrast, in mobile networks today the call-originator is charged.[10]

?? We need to be able to deal with inter-technology handovers[11]. In particular in a scenario with disconnected BANs (different administrative domains) with overlapping radio coverage, the availability of the BANs is only understood by the MN (as the independent networks do not know about each others existence) [A2.9]. Also, they may want to impose conflicting requirements on the MN (e.g. simultaneous 'handover now' messages). Clearly only the MN can decide how to react [A2.10].

?? The end-to-end argument (one of the Internet's architectural principles) implies that control and complexity should be on end systems, with the network as simple ("dumb") as possible, as discussed above.

However, the network has an important role in assisting or constraining handover. For example, the network might:

?? Suggest a handover to the MN, based on its own measurements (e.g. to load balance between cells) or its knowledge (e.g. a handover is imminent, because the MN is on a train)

?? Refuse a handover request or warning message from the MN, perhaps for its own policy or resource reasons

There may also be scenarios where the handover has to be network-controlled, because the MN is too simple to decide about a handover. Here a 'handover suggest' message from the network must be understood as compulsory by both the MN and the network. This case can be treated as a minor variant of the main mobile-controlled case.

A2.1.6.2.3 What handover performance does the user require?

The user may not be concerned about performance degradation during a handover, but in some cases will require:

[9] By contrast handovers in most existing mobile systems, such as GSM, are network-controlled.

[10] except for international roaming, where the international leg in the fixed network is billed to the call recipient. This is dealt with through a fixed policy subscription database.

[11] We define an "inter-technology handover" as: a handover during which the MN starts to use a different radio access technology. Such a handover may or may not cross a BAN boundary, and may even result in handover to a different network type (e.g. UMTS) altogether.

?? **Smooth Handover**
A handover that does not cause significant packet loss.

?? **Fast Handover**
A handover that does not cause significant packet delays.

?? **Seamless Handoff**
A fast and smooth handover.

In fact, in some ways it is misleading to talk about "handover performance' – actually the MN requires a particular 'QoS performance' (e.g. a maximum packet delay), regardless of whether it is camped on one cell or in the midst of handing over. Thus the Correspondent / Mobile Node needs to be told if its agreed QoS performance cannot be met, whatever the reason, and conversely it should not be told about a handover that does not break its QoS contract.

The network must therefore be in a position to deliver packets with whatever performance is required. Techniques that may be useful for achieving this during a handover are discussed in the next section.

A2.1.6.2.4 How can the network deliver this performance?

From a network perspective, we can distinguish two sorts of handover according to how it is initiated:

?? **Planned (expected) Handover**
This is the proactive case where some signalling can be done in advance of the MN being connected to the new BAR.

?? **Unplanned (unexpected) Handover**
This is the reactive case, where such signalling is not done in advance of the MN's move.

We require the network to support both sorts.

The planned case means that the network can take action before the MN's move and therefore we are more likely to maintain the MN's required QoS. Examples of action that the network could take in advance include:

?? Building a temporary tunnel from the old BAR to the new BAR. Packet loss in flight can then be obviated by forwarding packets down the tunnel to the NAR, and thence onto the MN as necessary.[12] [A2.11]

?? Transferring control information from the OAR to the NAR [A2.12] [A2.13]. A BAR controls how traffic is forwarded to and from a particular MN, for example through security keys (for encryption and message authentication, typically used at layer 2 but provided by layer 3), information supporting header [A2.14] and payload compression, multicast group membership details, and QoS infomation.

However, exactly the same functionality may also be needed by an unplanned handover. Therefore we believe that they should use a common set of messages and procedures. This will help to ensure consistency (a node reacts in the same way to the same message) and to guard against the failure of a message (e.g. if the planned handover messaging is not complete before the MN's move, then the network simply and intrinsically reverts to an unplanned handover), as well as reducing the amount of work for an implementer to do.

Handover is sometimes categorised according to whether it is "backward" (meaning it is initiated via the old BAR) or "forward" (i.e. initiated via the new BAR)[13]. This is primarily an implementation rather than a requirements issue and so is not considered further here.

We can also distinguish two sorts of handover according to the execution phase:

?? **Hard Handover**
The MN during a handover does not communicate simultaneously with the old and the new BAR (break-before-make).

?? **Soft Handover**
The MN can communicate simultaneously with the old and the new BAR (make-before-break)

[12] Either packet duplication or buffering is possible. This is an implementation issue and not considered here.

[13] In general backward handovers are planned and forward handovers are unplanned, but other combinations are possible, eg DECT handovers are planned but forward

Thus soft handover here refers to the idea of "IP diversity", where (in the downward direction) packets are sent from both BARs and the MN chooses the least errored on a packet-by-packet basis [14]. It ensures that handover is seamless, and may also help to overcome fading or interference problems. It requires a packet duplication function somewhere in the BAN.

It is an implementation issue whether soft handover needs to be supported as well as the basic hard handover. The decision will depend on the Layer 2 technology(s), the deployment scenario, the performance to be supported and so on.

In general, hard vs. soft and planned vs. unplanned handovers are independent. The most obvious combinations are that an unplanned handover is hard and a soft handover is planned, but the others are quite possible.

A2.1.6.3 Radio Resource Management

As in classical (2G/3G) terminology, the problem of radio resource management includes primarily the following subjects:

?? Deciding whether to allocate a (radio) channel to a terminal, taking into account current cell loading.

?? Carrying out the negotiation to allocate that channel to the terminal.

?? Deciding (on the basis of relative neighbour cell measurements, or in order to balance resources between different cells or within the resources of a single cell) to modify the radio channel allocated to a terminal – including as a special case handover.

Because of the interaction between network and radio-specific aspects, locating these problems cleanly within an overall design is difficult. They have to be supported in the overall BRAIN network. However, it is a design goal to maintain a clean separation between network related issues (which should be generic to any air interface) and issues specific to a given air interface. In BRAIN system, the network layer has to take part in any procedure that involves handovers between BARs (also called network layer handovers), since this requires re-routing and possibly network-related QoS reconfiguration within the wired access network. Therefore, these actions cannot take place entirely below the IP_2W interface, which also implies that the decision making activities must take place above this interface. This makes radio resource management at least partially the concern of the network layer.

We present here a model for decoupling air-interface specific aspects of radio resource management from the problem of handover. This architecture then forms part of the boundary between the generic network layer, and the air interface specific support:

?? The network layer provides information to the radio resource management (RRM) function, and responds to commands from it.

?? The RRM function operates air interface specific algorithms, which may also be tailored to a specific operational environment.

With this in mind, we can outline the following message flows at the MN and BAR as in Figure A2-9. Note that flows over the air interface are shown only as primitives at the IP_2W level (and in fact, only at the IP_2W control interface); these may be implemented over the air either as IP packet flows, or specialised layer 2 flows.

[14] This is different from UTRA's macrodiveristy (which is also sometimes called soft handover). It requires the transmissions from the two access points to be bit-aligned. It is expected that any such capability would be implemented at Layer 2, because it is too hard to do at the IP layer [15] (Note that this implies the two access points involved are attached to the same BAR, and incidentally therefore does not fall within our definition of soft handover)

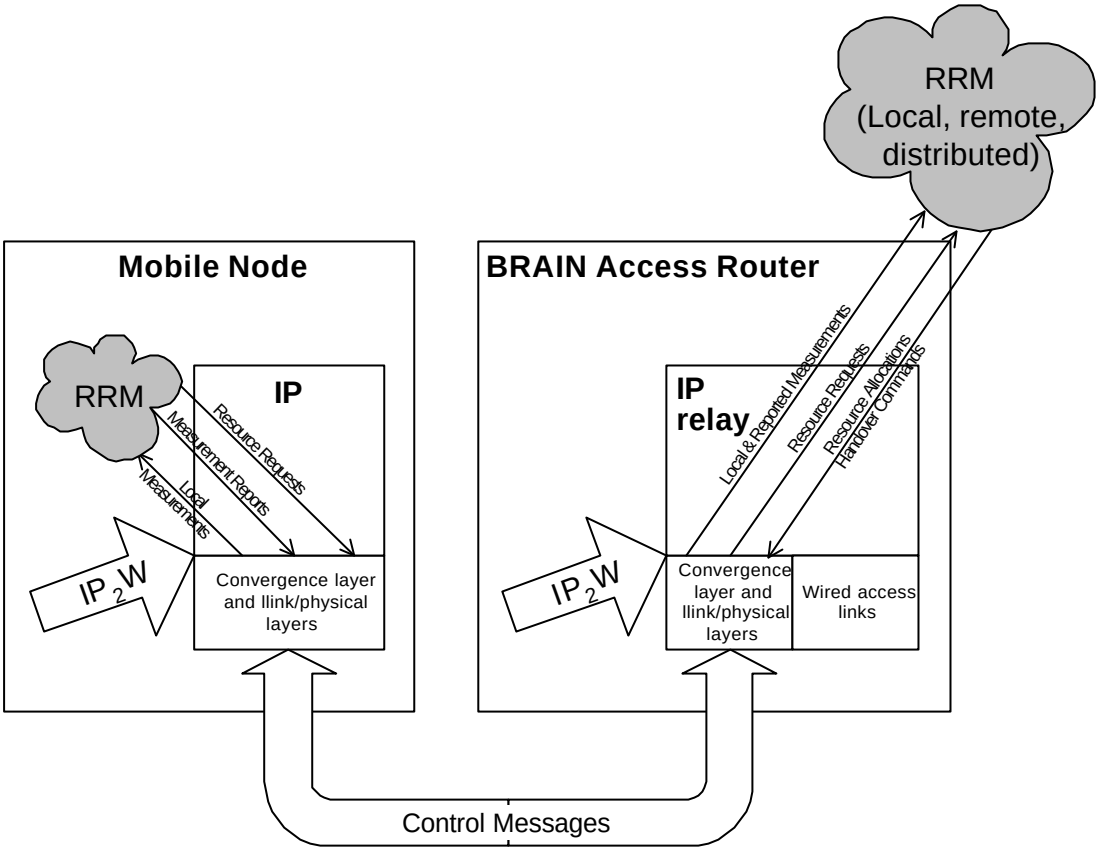


Figure A2-9: Radio Resource Management Message Exchanges

A2.1.7 Acknowledgements

This work has been performed in the framework of the IST project IST-1999-10050 BRAIN, which is partly funded by the European Union. The authors would like to acknowledge the contributions of their colleagues from Siemens AG, British Telecommunications PLC, Agora Systems S.A., Ericsson Radio Systems AB, France Télécom – R&D, INRIA, King's College London, Nokia Corporation, NTT DoCoMo, Sony International (Europe) GmbH, and T-Nova Deutsche Telekom Innovationsgesellschaft mbH.

This is an expanded version of a paper previously presented at the IST Mobile Summit (October 2000), which was co-authored with Hamid Aghvami (King's College), Markku Kojo (University of Helsinki) and Mika Liljeberg (Nokia Research Center) [A2.16].

A2.1.8 Paper References

[A2.1] Eardley P, Mihailovic M, Suihko T "A Framework for the Evaluation of IP Mobility Protocols", 11th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications. PIMRC 2000, 18-21 September 2000, IEEE, pp 451-7

[A2.2] B. Carpenter, "Architectural Principles of the Internet," RFC 1958, June 1996.

[A2.3] J.H. Saltzer, D.P. Reed, D.D. Clark, "End-To-End Arguments in System Design," ACM TOCS, Vol 2, Number 4, November 1984, pp 277-288.

[A2.4] David S. Isenberg, "The Dawn of the Stupid Network", ACM Networker 2.1, February/March 1998, pp. 24-31.

[A2.5] IETF draft "Mobile IP Authentication, Authorization, and Accounting Requirements", draft-ietf-mobileip-aaa-reqs-03.txt"

[A2.6] RADIUS C. Rigney, S. Willens, A. Rubens and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", RFC2865, June 2000.

[A2.7] P. Calhoun, A. Rubens, H. Akhtar and E. Guttman, "DIAMETER Base Protocol", draft-calhoun-diameter-16.txt, Internet Draft (work in progress), July 2000. http://www.diameter.org /

[A2.8] C. Keszei, J. Manner, Z. Turányi, A. Valkó, "Mobility Management and QoS in BRAIN Access Networks", 1st International Workshop on Broadband radio access for IP based networks, 20 November 2000

[A2.9] P. Conforto, G. Losquadro, C. Tocci, M. Luglio, R.E. Sheriff, "SUITED/GMBS system architecture", IST Mobile Communications Summit 2000, 1-4 October 2000, pp 115-121

[A2.10] A. O'Neill, G. Tsirtsis, S. Corson, "Generalized IP Handoff", Internet-Draft (work in progress), August 2000, draft-oneill-craps-handoff-00.txt

[A2.11] G. Krishnamurthi, R. Chalmers, C. Perkins, "Buffer management for smooth handovers in Mobile IPv6", Internet-Draft (work in progress), 13 July 2000, draft-krishnamurthi-mobileip-buffer6-oo.txt

[A2.12] R. Koodli, C. Perkins, "A framework for smooth handovers with Mobile IPv6", Internet-Draft (work in progress), 13 July 2000, draft-koodli-mobileip-smoothv6-00.txt

[A2.13] A. O'Neill, G. Tsirtsis, S. Corson, "State transfer between access routes during handoff", Internet-Draft (work in progress), August 2000, draft-oneill-handoff-state-00.txt

[A2.14] R. Koodli, M. Tiwari, C. Perkins, "Header compression state relocation in IP Mobile networks", Internet-Draft (work in progress), 13 July 2000, draft-koodli-rohc-hc-relocate-00.txt

[A2.15] J. Kempf, P. McCann, P.Roberts, "IP Mobility and the CDMA Radio Access Network: Applicability Statement for Soft Handoff", IETF draft-kempf-cdma-appl-00.txt, July 2000.

[A2.16] R. Hancock, H. Aghvami, M. Kojo and M. Liljeberg, "The architecture of the BRAIN network layer", IST Mobile Communications Summit 2000, 1-4 October 2000, pp 581-586

A2.2 Mobility Related Terminology

This section presents the IETF draft (work in progress) that has been submitted to the IETF Seamoby working group.

A2.2.1 IETF Draft

Internet Engineering Task Force
Internet-Draft
Expires: September, 2001

J. Manner
M. Kojo
University of Helsinki
T. Suihko
VTT Information Technology
P. Eardley
D. Wisely
BT
R. Hancock
Siemens/Roke Manor Research
Nikos Georganopoulos
King's College London
March 2, 2001

Mobility Related Terminology
<draft-manner-seamoby-terms-01.txt>

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at: http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at: http://www.ietf.org/shadow.html.

This Internet-Draft will expire in September, 2001.

Abstract

There is a need for common definitions of terminology in the work tobe done around IP mobility. This memo defines terms for mobility related terminology. It is intended as a living document for use by the Seamoby working group, and especially for use in Seamoby drafts and in WG discussions.

Table of Contents

1 Introduction 2
2 Definitions 3
2.1 Network Components..... 3
2.2 Reference Architecture..... 5
2.3 Handover Terminology 6
2.3.1 Scope of Handover..... 6
2.3.2 Technologies and Network Interfaces 6
2.3.3 Handover Control 7
2.3.4 Simultaneous connectivity to Access Routers 8
2.3.5 Performance and Functional Aspects 8
2.4 Micro diversity, Macro diversity, and IP diversity 9
2.5 Mobile Host States and Modes 9
2.6 User, Personal and Host Mobility 10
2.7 Macro and Micro Mobility 11
3 Acknowledgement 11
4 References 12
5 Author's Addresses 13
6 Appendix A - Examples 15

1. Introduction

This document presents a terminology to be used for documents and discussions within the Seamoby Working Group. Other working groups may also take advantage of this terminology in order to create a common terminology for the area of mobility.

Some terms and their definitions that are not directly related to the IP world are included for the purpose of harmonizing the terminology, for example, 'Access Point' and 'base station' refer to the same component but 'Access Router' has a very different meaning. The presented terminology may not be adequate to cover mobile ad-hoc networks.

The proposed terminology is not meant to 'push' new terminology. Rather the authors would welcome discussion on more exact definitions as well as missing or unnecessary terms. This work is a collaborative enterprise between people from many different engineering backgrounds and so already presents a first step in harmonizing the terminology.

2. Definitions

2.1. Network Components

Note: The fundamental new concept to be introduced is that of the Access Network (AN) which supports enhanced mobility. It is a working assumption that to support routing and QoS for mobile nodes, we need specialized routing functions (i.e. not OSPF or other standard IGPs) which are used to maintain forwarding information for these mobile nodes as they change their points of attachment to the Access Network, and these functions are implemented in IP routers with this additional capability. We can distinguish three types of Access Network components: Access Routers (AR) which handle the last hop to the mobile; Access Network Gateways (ANG) which form the boundary on the fixed network side and shield the fixed network from the specialized routing protocols; and (optionally) other internal Access Network Routers which may also be needed in some cases to support the protocols. The Access Network consists of the equipment needed to support this specialized routing, i.e. AR/ANG/ANR.

Mobile Node (MN)

An IP node capable of changing its point of attachment to the network. The Mobile Node may have routing functionality.

Mobile Host (MH)

A mobile node that is an end host. In this document we use the term Mobile Host, although the term Mobile Node could be used in most, if not all, cases where a Mobile Node serves as a mobile router of a mobile network.

Access Link (AL)

A last-hop link between a Mobile Host and an Access Router. That is, a facility or medium over which an Access Point and a layer 2 wireless device attached to the Mobile Host can communicate at the link layer, i.e., the layer immediately below IP. The wireless device may be co-located with the Mobile Host.

Access Point (AP)

An Access Point is a layer 2 device which is connected to one or more Access Routers and offers the wireless link connection to the Mobile Host. Access Points are sometimes called ³base stations³ or ³access point transceivers³. An Access Point may be a separate entity or co-located with an Access Router.

Radio Cell

The geographical area within which an Access Point provides radio coverage, i.e. where radio communication between a Mobile Host and the specific Access Point is possible. Adapted from [A2.21].

Access Network Router (ANR)

An IP router in the Access Network. An Access Network Router may include Access Network specific functionalities, for example, on mobility and/or QoS. This is to distinguish between ordinary routers and routers that have Access Network-related special functionality.

Access Router (AR)

An Access Network Router residing on the edge of an Access Network and connected to one or more Access Points. The Access Points may be of different technology. An Access Router offers IP connectivity to Mobile Hosts, acting as a default router to the Mobile Hosts it is currently serving. The Access Router may include intelligence beyond a simple forwarding service offered by ordinary IP routers.

Access Network Gateway (ANG)

An Access Network Router that separates an Access Network from other IP networks. An Access Router and an Access Network Gateway may be the same physical node. The Access Network Gateway looks to the other IP networks like a standard IP router.

Access Network (AN)

An IP network which includes one or more Access Network Routers.

Administrative Domain(AD)

A collection of networks under the same administrative control and grouped together for administrative purposes. [A2.26]

Serving Access Router (SAR)

The Access Router currently offering the connectivity to the Mobile Host. This is usually the point of departure for the Mobile Host as it makes its way towards a new Access Router (then Serving Access Router takes the role of the Old Access Router). There may be several Serving Access Routers serving the Mobile Host at the same time.

Old Access Router (OAR)

An Access Router that offered connectivity to the Mobile Host prior to a handover. This is the Serving Access Router that will cease or has ceased to offer connectivity to the Mobile Host.

New Access Router (NAR)

The Access Router that offers connectivity to the Mobile Host after a handover.

Candidate Access Router (CAR)

An Access Router to which the Mobile Host may move next. A handover scheme may support several Candidate Access Routers.

2.2. Reference Architecture

The following figure (Fig. 1) presents a reference architecture to illustrate the presented network components. The figure presents two examples of possible AN topologies.

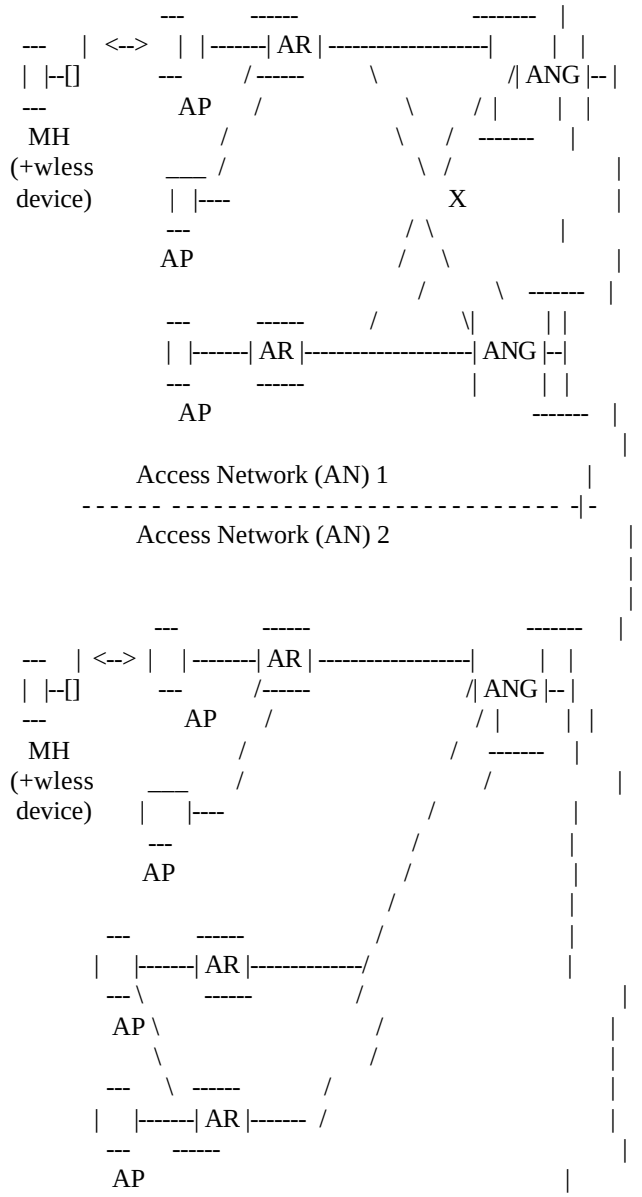


Figure 1: Reference Network Architecture

2.3. Handover Terminology

These terms refer to different approaches to supporting different aspects of mobility.

- Roaming refers to a particular aspect of user mobility. Roaming is an operator-based term involving formal agreements between operators that allows a mobile to get connectivity from a foreign network. Roaming includes, for example, the functionality by which users can communicate their identity to the local AN so that inter-AN agreements can be activated and service and applications in the MH's home network can be made available to the user locally.

- Handover (also known as handoff) is the process involved when an active MH (in the Active State, see section 2.4) changes its point of attachment to the network, or when such a change is attempted. The access network may provide particular capabilities to minimize the interruption to sessions in progress.

There are different types of handover classified according to different aspects involved in the handover. Some of this terminology follows the description of [A2.22].

2.3.1. Scope of Handover

- Layer 2 Handover: When a MH changes APs (or some other aspect of the radio channel) connected to the same AR's interface then a layer 2 handover occurs. This type of handover is transparent to the routing at the IP layer (or it appears simply as a link layer reconfiguration without any mobility implications).

- Intra-AR Handover: This is a handover which changes the AR's IP layer's network interface to the mobile. This causes routing changes internal to the AR. The IP address by which the MH is reachable does not change.

- Intra-AN Handover: When the MH changes ARs inside the same AN then this handover occurs. Such a handover is not necessarily visible outside the AN. In case the ANG serving the MH changes, this handover is seen outside the AN due to a change in the routing paths. The IP address by which the MH is reachable does not change. Note that the ANG may change for only some of the MH's data flows.

- Inter-AN Handover: When the MH moves to a new AN then this handover occurs. This requires some sort of host mobility across ANs, which has to be provided by the external IP core. Note that this would have to involve the assignment of a new IP address to the MH.

2.3.2. Technologies and Network Interfaces

- Intra-technology Handover: A handover between equipment of the same technology.

- Inter-technology Handover: A handover between equipment of different technologies.

- Horizontal Handover: from the IP point of view a horizontal handover happens if the MH communicates with the AN via the same network interface; the network interface is the same before and after the handover. A horizontal handover is typically also an intra- technology handover but it can be an inter-technology handover if the layer 2 device attached to the MH can do a layer 2 handover between two different technologies without changing the network interface seen by the IP layer.

- Vertical Handover: in a vertical handover the MH's network interface to the Access Network changes. A vertical handover is typically an inter-technology handover but it may also be an intra- technology handover if the MH has several network interfaces of the same type. That is, after the handover, the IP layer communicates with the AN through a different network interface.

The different handover types defined in this section and in section 2.2.1 have no direct relationship. In particular, a MH can do an intra-AN handover of any of the types defined above.

Note that the horizontal and vertical handovers are not tied to a change in the link layer technology. They define whether, after a handover, the IP packet flow goes through the same (horizontal handover) or a different (vertical handover) network interface. These two handovers neither define whether the AR changes as a result of a handover.

2.3.3. Handover Control

A handover must be one of the following two types (a):

- Mobile-initiated Handover: the MH is the one that makes the initial decision to initiate the handover.

- Network-initiated Handover: the network makes the initial decision to initiate the handover.

A handover is also one of the following two types (b):

- Mobile-controlled Handover (MCHO): the MH has the primary control over the handover process.

- Network-controlled Handover (NCHO): the network has the primary control over the handover process.

A handover may also be either of these two types (c):

- Mobile-assisted handover: information and measurement from the MH are used to decide on the execution of a handover.

- Network-assisted handover: a handover where the AN collects information that can be used in a handover decision.

A handover is also one of the following two types (d):

- Backward handover: a handover either initiated by the OAR, or where the MH initiates a handover via the OAR.

- Forward handover: a handover either initiated by the NAR, or where the MH initiates a handover via the NAR.

The handover is also either proactive or reactive (e):

- Planned handover: a proactive (expected) handover where some signalling can be done in advance of the MH getting connected to the new AR, e.g. building a temporary tunnel from the old AR to the new AR.

- Unplanned handover: a reactive (unexpected) handover, where no signalling is done in advance of the MH's move of the OAR to the new AR.

The five handover types (a-e) are orthogonal. Type 'c' may be present in a handover, the other types are always present.

2.3.4. Simultaneous connectivity to Access Routers

- Make-before-break handover (MBB): During a MBB handover the MH can communicate simultaneously with the old and new AR. This should not be confused with "soft handover" which relies on macro diversity.

- Break-before-make handover (BBM): During a BBM handover the MH does not communicate simultaneously with the old and the new AR.

2.3.5. Performance and Functional Aspects

- Handover Latency: Handover latency is the time difference between when a MH is last able to send and/or receive an IP packet by way of the OAR, until when the MH is able to send and/or receive an IP packet through the NAR. Adapted from [A2.22]

- Smooth handover: A handover that aims primarily to minimize packet loss, with no explicit concern for additional delays in packet forwarding.

- Fast handover: A handover that aims primarily to minimize delay, with no explicit interest in packet loss.

- Seamless handover: The absolute reference definition for a seamless handover is one in which there is no change in service capability, security, or quality. In practice, some degradation in service is to be expected. The definition of a seamless handover in the practical

case should be that other protocols, applications, or end users do not detect any change in service capability, security or quality, which would have a bearing on their (normal) operation. See [A2.21] for more discussion on the topic.

2.4. Micro diversity, Macro diversity, and IP diversity

Certain air interfaces (e.g. UTRAN FDD mode) require or at least support the concepts of macro diversity combining. Essentially, this refers to the fact that a single MH is able to send and receive over two independent radio channels ('diversity branches') at the same time; the information received over different branches is compared and that from the better branch passed to the upper layers. This can be used both to improve overall performance, and to provide a seamless type of handover at layer 2, since a new branch can be added before the old is deleted. See also [A2.20].

It is necessary to differentiate between combining/diversity that occurs layer 1/2 (physical and radio link layers) where the relevant unit of data is the radio frame, and that which occurs at layer 3, the network layer, where what is considered is the IP packet itself.

In the following definitions micro- and macro diversity refer to L1/L2 and IP diversity refers to L3.

- Micro diversity is the term used for the case where, for example, two antennas on the same transmitter send the same signal to a receiver over a slightly different path to overcome fading.

- Macro diversity takes place when the duplicating / combining actions take place over multiple APs, possibly attached to different ARs. This may require support from the network layer to move the radio frames between the base stations and a central combining point.

- IP diversity means the splitting and combining of packets at the IP level.

2.5. Mobile Host States and Modes

Mobile systems may employ the use of MH states in order to operate more efficiently without degrading the performance of the system. The term 'mode' is also common and means the same as 'state'.

A MH is always in one of the following three states:

- Active State is when the AN knows the MH's SAR and the MH can send and receive IP packets. The AL may not be active, but the radio layer is able to establish one without assistance from the network layer. The MH has an IP address assigned.

- Idle State is when the AN knows the MH's Paging Area, but the MH has no SAR and so packets cannot be delivered to the MH without the

AN initiating paging.

- Detached State is when the MH is in neither the Active nor Idle State. The MH does not have an IP address from the AN.

- Paging is a procedure initiated by the Access Network to move an Idle MH into the Active State. As a result of paging, the MH establishes a SAR and the IP routes are set up.

- Location updating is a procedure initiated by the MH, by which it informs the AN that it has moved into a new paging area.

- A Paging Area is a part of the Access Network, typically containing a number of ARs/APs, which corresponds to some geographical area. The AN keeps and updates a list of all the Idle MHs present in the area.
If the MH is within the radio coverage of the area it will be able to receive paging messages sent within that Paging Area.

Note: in fact, as well as the MH being in one of these three states, the AN also stores which state it believes the MH is in. Normally these are consistent; the definitions above assume so.

2.6. User, Personal and Host Mobility

Different sorts of mobility management may be required of a mobile system. We can differentiate between user, personal and host mobility.

- User mobility: refers to the ability of a user to access services from different physical hosts. This usually means, the user has an account on these different hosts or that a host does not restrict users from using the host to access services.

- Personal mobility: complements user mobility with the ability to track the user's location and provide the users current location to allow sessions to be initiated by and towards the user by anyone on any other network. Personal mobility is also concerned with enabling associated security, billing and service subscription authorization made between administrative domains.

- Host mobility: refers to the function of allowing a mobile host to change its point of attachment to the network, without interrupting IP packet delivery to/from that host. There may be different sub-functions depending on what the current level of service is being provided; in particular, support for host mobility usually implies active and idle modes of operation, depending on whether the host has any current sessions or not. Access Network procedures are required to keep track of the current point of attachment of all the MHs or establish it at will. Accurate location and routing procedures are required in order to maintain the integrity of the communication. Host mobility is often called 'terminal mobility'.

2.7. Macro and Micro Mobility

Macro and micro mobility refer to host mobility in wide and local geographical area. Correspondingly, macro- and micro-mobility management refer to the scope of protocol operations in mobility management.

- Macro mobility refers literally to 'mobility over a large area'. This includes mobility support and associated address registration procedures that are needed when a mobile host moves between IP domains. Inter-AN handovers typically involve macro-mobility protocols. Mobile-IP can be seen as a means to provide macro mobility.

- Micro mobility refers to 'mobility over a small area'. Usually this means mobility within an IP domain with an emphasis on support for active mode using handover, although it may include idle mode procedures also. Micro-mobility protocols exploit the locality of movement by confining movement related changes and signalling to the access network.

3. Acknowledgement

This work has been performed in the framework of the IST project IST-1999-10050 BRAIN, which is partly funded by the European Union. The authors would like to acknowledge the contributions of their colleagues from Siemens AG, British Telecommunications PLC, Agora Systems S.A., Ericsson Radio Systems AB, France Telecom R&D, INRIA, King's College London, Nokia Corporation, NTT DoCoMo, Sony International (Europe) GmbH, and T-Nova Deutsche Telekom Innovations-gesellschaft GmbH.

Some definitions of terminology have been adapted from [A2.17], [A2.18], [A2.19], [A2.22], [A2.24], [A2.25].

4. References

[A2.17] Blair, D., Tweedly, A., Thomas, M., Trostle, J., Ramalho, M., "Realtime Mobile IPv6 Framework". Internet Draft (work in progress), November 2000 (draft-blair-rt-mobileipv6-seamoby-00.txt).

[A2.18] Deering, S., Hinden, R., "Internet Protocol, Version 6 (IPv6) Specification". Internet Engineering Task Force, Request for Comments (RFC) 2460, December 1998.

[A2.19] Gustafsson, E., Jonsson, A., Perkins, C., "Mobile IP Regional Registration". Internet Draft (work in progress), July 2000 (draft-ietf-mobileip-reg-tunnel-03.txt).

[A2.20] Kempf, J., McCann, P., Roberts, P., "IP Mobility and the CDMA Radio Access Network: Applicability Statement for Soft Handoff", Internet Draft (work in progress), July 2000 (draft-kempf-cdma-appl-00.txt).

[A2.21] Levkowetz, O.H. et. al., "Problem Description: Reasons For Doing Context Transfers Between Nodes in an IP Access Network. Internet Draft (work in progress), February 2001(draft-ietf-seamoby-context-transfer-problem-stat-00.txt).

[A2.22] MIPv6 Handover Design Team, "Fast Handovers for Mobile IPv6". Internet Draft (work in progress), February 2001 (draft-designteam-fast-mipv6-01.txt).

[A2.23] Pandya, R., "Emerging mobile and personal communication systems," IEEE Communications Magazine , vol. 33, pp. 44--52, June 1995.

[A2.24] Perkins, C., "IP Mobility Support". Internet Engineering Task Force, Request for Comments (RFC) 2002, October 1996.

[A2.25] Ramjee, R., La Porta, T., Thuel, S., Varadhan, K., Salgarelli, L., "IP micro-mobility support using HAWAII". Internet Draft (work in progress), July 2000 (draft-ietf-mobileip-hawaii-01.txt).

[A2.26] Yavatkar, et al., "A Framework for Policy-based Admission Control", RFC 2753, January 2000.

5. Author's Addresses

Questions about this document may be directed to:

Jukka Manner
Department of Computer Science
University of Helsinki
P.O. Box 26 (Teollisuuskatu 23)
FIN-00014 HELSINKI
Finland

Voice: +358-9-191-44210
Fax: +358-9-191-44441
E-Mail: jmanner@cs.helsinki.fi

Markku Kojo
Department of Computer Science
University of Helsinki
P.O. Box 26 (Teollisuuskatu 23)
FIN-00014 HELSINKI
Finland

Voice: +358-9-191-44179
Fax: +358-9-191-44441
E-Mail: kojo@cs.helsinki.fi

Tapio Suihko
VTT Information Technology
P.O. Box 1203
FIN-02044 VTT
Finland

Voice: +358-9-456-6078
Fax: +358-9-456-7028
E-Mail: tapio.suihko@vtt.fi

Phil Eardley
BTexaCT
Adastral Park
Martlesham
Ipswich IP5 3RE
United Kingdom

Voice: +44-1473-645938
Fax: +44-1473-646885
E-Mail: philip.eardley@bt.com

Dave Wisely
BTexaCT
Adastral Park
Martlesham
Ipswich IP5 3RE
United Kingdom

Voice: +44-1473-643848
Fax: +44-1473-646885
E-Mail: dave.wisely@bt.com

Robert Hancock
Roke Manor Research Ltd
Romsey, Hants, SO51 0ZN
United Kingdom

Voice: +44-1794-833601
Fax: +44-1794-833434
E-Mail: robert.hancock@roke.co.uk

Nikos Georganopoulos
King's College London
Strand
London WC2R 2LS
United Kingdom

Voice: +44-20-78482889
Fax: +44-20-78482664
E-Mail: nikolaos.georganopoulos@kcl.ac.uk)

6. Appendix A - Examples

This appendix provides examples for the terminology presented.

A.1 Mobility

Host mobility is logically independent of user mobility, although in real networks, at least the address management functions are often required to attach the host to the network in the first place. In addition, if the network wishes to determine whether access is authorized (and if so, who to charge for it), then this may be tied to the identity of the user of the terminal.

An example of user mobility would be a campus network, where a student can log into the campus network from several workstations and still get his/her files, emails, etc. services automatically.

Personal mobility support typically amounts to the maintenance and update of some sort of address mapping database, such as a SIP server or DNS server; it is also possible for the personal mobility support function to take a part in forwarding control messages between end user and correspondent rather than simply acting as a database. SIP is a protocol for session initiation in IP networks. It includes registration procedures which partially support personal mobility (namely, the ability for the network to route a session towards a user at a local IP address).

Personal mobility has been defined in [A2.23] as "the ability of end users to originate and receive calls and access subscribed telecommunication services on any terminal in any location, and the ability of the network to identify end users as they move. Personal mobility is based on the use of a unique personal identity (i.e., personal number)."

Roaming, in its original (GSM) sense, is the ability of a user to connect to the networks owned by operators other than the one he has a direct formal relationship with. More recently (e.g. in data networks and UMTS) it also refers to the fact that the 'foreign' network may still be able to provide user-customized services, e.g. QoS profiles for specific applications.

HAWAII, Cellular IP, Regional Registration and EMA are examples of micro mobility schemes, with the assumption that Mobile IP is used for macro mobility.

WLAN technologies such as IEEE 802.11 typically support aspects of user and host mobility in a minimal way. User mobility procedures (for access control and so on) are defined only over the air interface (and the way these are handled within the network is not further defined).

PLMNs (GSM/UMTS) typically have extensive support for both user and host mobility. Complete sets of protocols (both over the air and on

the network side) are provided for user mobility, including customized service provision. Handover for host mobility is also supported, both within access networks, and also within the GSM/UMTS core network for mobility between access networks of the same operator.

A.2 Handovers

A hard handover is required where a MH is not able to receive or send traffic from/to two APs simultaneously. In order to move the traffic channel from the old to the new access point the MH abruptly changes the frequency/timeslot/code on which it is transmitting and listening to new values associated with a new access point.

A good example of hard handover is GSM where the mobile listens for new base stations, reports back to the network the signal strength and identity of the new base station(s) heard. When the old base station decides that a handover is required it instructs the new base station to set up resources and, when confirmed, instructs the mobile to switch to a new frequency and time slot. This sort of hand over is called hard, mobile assisted, network initiated and backward (meaning that the old base station is responsible for handling the change-over).

In a TDMA system, such as GSM, the hard hand over is delayed until the mobile has moved well within the coverage of the new base station. If the handover threshold was set to the point where the new base station signal exceeded the old then there would be a very large number of handovers as the mobile moved through the region between the cells and radio signals fluctuated, this would create a large signalling traffic. To avoid this a large hysteresis is set, i.e. the new base station must be (say) 10dB stronger for handover to occur. If the same was done in W-CDMA then the mobile would be transmitting a powerful signal to the old base station and creating interference for other users, since in CDMA everyone else's transmissions are seen as noise, thus reducing capacity. To avoid this soft handover is used, giving an estimated doubling in capacity.

Support for soft handover (in a single mode terminal) is characteristic of radio interfaces which also require macro diversity (bi-casting) for interference limitation but the two concepts are logically independent.

A good example of soft handover is the UTRAN FDD mode. W-CDMA is particularly suited to soft handover because of the design of the receivers and transmitters: typically a rake receiver will be used to overcome the multi-path fading of the wide-band channel. Rake receivers have a number of so-called fingers, each effectively separate detectors, that are tuned to the same signal (e.g. spreading code) but delayed by different times. When the delay times are correctly adjusted and the various components properly combined (this is micro diversity combining) the effect of multi-path fading is removed. The rake receiver can also be used to detect signals from

different transmitters by tuning the fingers to different spreading codes. Soft handover is used in UTRAN FDD mode to also increase capacity.

Every handover can be seen as Context-aware Handovers. In PLMNs the context to be fulfilled is that the new AP can accommodate the new mobile, for example, the new GSM cell can serve the incoming phone. Lately, the notion of Context-aware Handovers has been enlarged by, for example, QoS-aware handovers, meaning that the handover is governed by the need to support the QoS-context of the moving mobile in order to keep the service level assured to the user of the MH.

A.3 Diversity combining

In the case of UMTS it is radio frames that are duplicated at some point in the network (the serving RNC) and sent to a number of Node Bs and, possibly via other (drift) RNCs. The combining that takes place at the serving RNC in the uplink direction is typically based on some simple quality comparison of the various received frames, which implies that the various copies of these frames must contain identical upper layer information. The serving RNC also has to do buffering to take account of the differing time of flight from each Node B to the RNC.

A.4 Miscellaneous

In a GPRS/UMTS system the Access Network Gateway node would be the GGSN component. The ANG can provide support for mobility of hosts, admission control, policy enforcement, and Foreign Agent functionality.

When presenting a mobile network topology, APs and ARs are usually pictured as separate components. This is the case with GSM/GPRS/UMTS presentations, for example. From the IP point of view APs are not directly visible. An AP should only be seen from the MH's or AR's IP layer as a link (interface) connecting MHs to the AR.

When the mobile moves through the network, depending on the mobility mechanism, the OAR will forward packets destined to the old MHs address to the SAR which currently serves the MH. At the same time the handover mechanism may be studying CARs to find the best NAR where the MH will be handed next.

Note that when a network includes IP-over-IP tunnels, we need to be very careful about which IP routing and IP address we are discussing.

Full Copyright Statement

A2.3 BRAIN – UMTS Interoperation

In this section of the annex we will look in much more detail at that summarised in the similarly titled section of the short report. The layout of this more detailed look will attempt to follow that of the short report but provide some backup for the statements made. All references to handover in this section of the report refer to vertical handover between two networks.

A2.3.1 Coupling

We have identified three types of coupling that can be used; 'no-coupling', 'loose coupling' and 'tight coupling' there are a number of variations that can be applied in the their implementation, most noticeably in the tight-coupling approach.

A2.3.1.1 No Coupling

This approach assumes that the operators of the UMTS and BRAIN networks share no resources at all to accomplish a handover some higher level network functionality is required this could be provided by a third operator/service provider or be provided by one of the two network operators. This higher level network functionality could be an implementation of MIP HA or a SIP based solution. Shown in Figure A2-10is an overview of how this architecture would look.

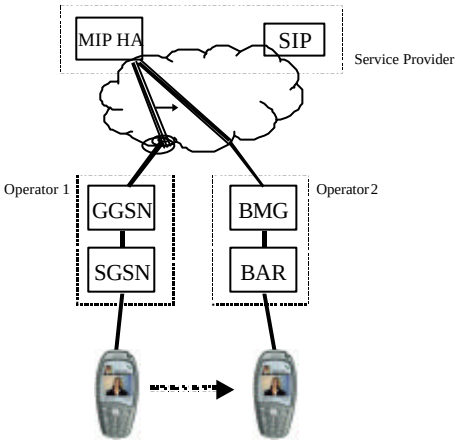


Figure A2-10: The No Coupling Architecture.

The no coupling approach will demonstrate a slower handover as path update and control messages will need to propagate to higher levels in the network, then in the other coupling approaches. The advantage of the no coupling approach is its flexibility and independence from a single operator. A subscriber can choose who provides each part of the service portfolio they have.

Lets consider an example of how the SIP based solution could provide for terminal mobility:

A2.3.1.1.1 An Example of SIP Terminal Mobility in a no-coupling architecture.

Using SIP at a higher layer in the network structure can provide terminal mobility [A2.27]. This is achievable when you consider Terminal Mobility to only be a special case of personal mobility. Terminal Mobility could occur at two different times and the solutions for this can be different.

Out of call terminal mobility is achieved by the standard use of SIP and a redirect or proxy server. When the terminal moves it updates the registry of its SIP proxy or redirect server with its new location. It public address is that which the SIP server recognises and redirects the Caller to the new location of the user.

In call terminal mobility is achieved by sending an invite message to the CN with the updated contact and session description. This can be seen in Figure A2-11.

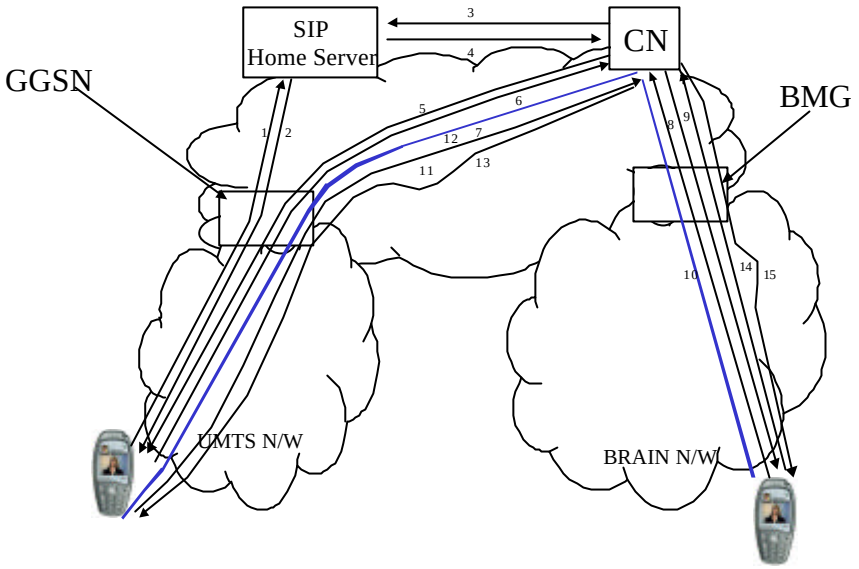


Figure A2-11: Call flow for SIP terminal mobility during HO between UMTS & BRAIN networks.

Lets assume the terminal is registered with both a BRAIN and a UMTS network. The MN registers its current location with its SIP server (1) which acknowledges this (2). When the CN calls the user {INVITE: SIP:david.higgins@sip.com} (3) the SIP server respond with the current location {302: Moved temporarily Contact: SIP:david.higgins@umts.com} (4). The CN sends a new invite {INVITE: SIP:david.higgins@umts.com} (5), the MN acknowledges {200: OK} (6) and communications can take place (7). The MN now wants to use the BRAIN network it send a new invite to the CN with the new contact and session description {INVITE: SIP: CN@other.net Contact: SIP:david.higgins@brain.net} (8) CN acknowledges {200: OK} (9) and communications on the new link can occur (10). The MN can close the old connection {Bye} (11) the link stops (12) and the CN acknowledges {200: OK}. When they have finished the MN send Bye on the new link (14) which again is acknowledged (15). The actual SIP messages have been abbreviated in this example but the essence is there.

A2.3.1.2 Loose Coupling

In the loose coupling scheme the operators of both networks could still be different however use of the same AAA subscriber databases for both networks occurs e.g. for functions such as billing and management. This form of coupling can exhibit the same problems in that the information during handover needs to be passed to high levels of the network thus incurring time/speed penalties. The integration of billing etc however provides significant advantages to both the operators and the subscriber. The subscriber would get only one bill outlining the costs for all access and would also get a consistent type of customer service. For the operator the advantages are less equipment, reduced costs and the retention of customers. This however is why in this case the operators are likely to the same of both networks or trusted partners, those with whom, close working is already in place.

The loose coupling is mainly related to security and there are two basic approaches to integrating the BRAIN security requirements with the 3G system. One is to map the BRAIN authentication message exchanges that would take place over the air interface (as defined by IP2W) onto equivalent messages that would typically be used to exploit the presence of a SIM/USIM. This requires that the authentication protocol is sufficiently generic to support these messages and ties the strength of the authentication to the quality of the 2/3G security architecture and implementation, on the other hand, it does provide the best integration with the HLR, although the benefit only arises when the same SIM/USIM is used in all the terminals belonging to the subscriber concerned. The other method allows the use of any authentication protocol, typically an extensible protocol such as PPP EAP or IEEE 802.1x, and links the security exchange directly to the operator's authentication and billing system, typically using a backhaul protocol such as RADIUS to do so. Here, the level of integration is lower, but there is more flexibility in deployment and reduced risk of incompatibility between the authentication exchange message sequence and the requirements of the SIM/USIM system.

However keeping in sight the experience of the user, with the exception of inter-system handover performance, the loose coupling approach would be able to add the same type of IP multimedia applications to the end user as any of the tight coupling applications. The technical difference is that these IP multimedia applications will not be built on the standard UMTS bearer services, but something else (whatever bearer services the QoS team eventually choose to provide); however, this difference should not be visible to the end user. The practical difference is therefore that HIPERLAN/2 - UMTS handover will be less smooth, and joint management of the radio resource by the operator will be harder.

A2.3.1.3 Tight Coupling.

When considering tight coupling when the two networks are integrated such as to share part of the core network resources, in UMTS there are two levels at which this can occur: Access via the GGSN and access via the SGSN shown in Figure A2-12.

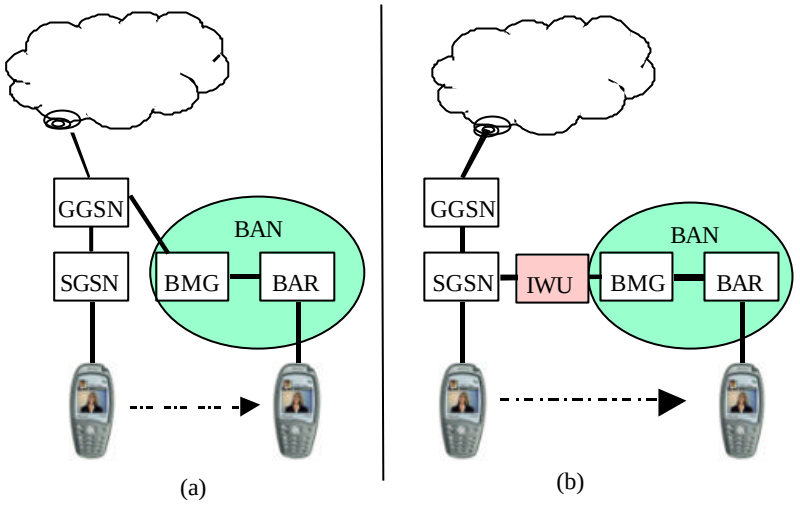


Figure A2-12: The Tight Coupling Architecture (a) Access via the GGSN; (b) Access via the SGSN.

It is important to consider that with tight coupling there will be affects on the core network. One of these is that the BRAIN network will support considerably higher bit rate services then would be expected on the UMTS network therefore the core network and in the two proposed methods of tight coupling GGSN or GGSN & SGSN respectively would need to be uprated to be able to cope with the increased demands placed upon them.

The interworking between BRAIN and UMTS shall not put any new requirements on the network to MN interface for pure BRAIN terminals. This means that full interworking is needed between UMTS and BRAIN protocol (control and user data protocols) in case (b) above. With a tight integration between GGSN and BMG this interworking can be simplified (see further below).

Let's consider these two cases of tight coupling.

A2.3.1.3.1 Tight Coupling (access via GGSN)

As can be seen in Figure A2-12 (a) one way of tight coupling is for the BRAIN network to have access via the GGSN. This means that UMTS and BRAIN micro mobility as well as internal protocols can be kept separately. The advantage of tight coupling must be to reduce the handover time, as propagation of update messages does not need to go as far up the network hierarchy. The network from the GGSN must be able to sustain the higher bit rates expected from the BRAIN access technologies.

The Figure A2-13 indicates a possible solution. Observe that a tight integration between BMG and GGSN as well as between AAA and HLR can be possible and thereby simplifying the interworking.

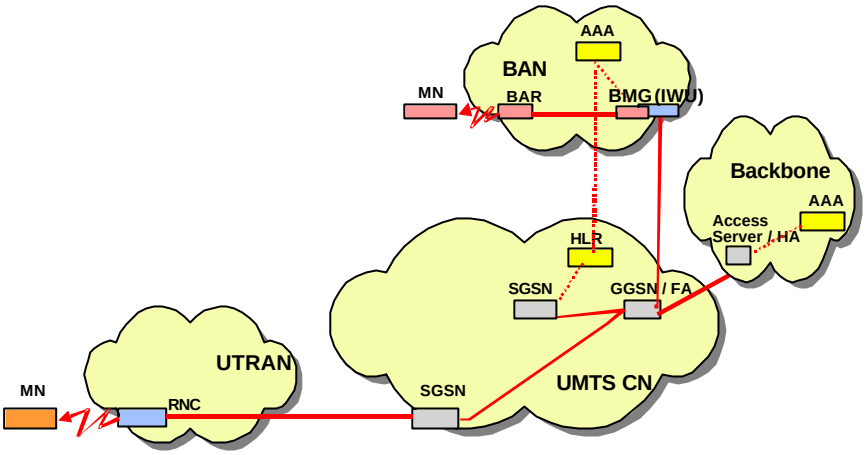


Figure A2-13 Tight Integration at GGSN and HLR level

A2.3.1.3.2 Tight Coupling (access via SGSN)

In this case shown in Figure A2-12(b) and Figure A2-14 the tight coupling is achieved by the BRAIN network having access via the SGSN, this means the BAN is effectively a new UMTS Access network. This implicates an Iu interface is required at the output of the BMG this will necessitate an IWU.

The interworking function (IWU) shown would be co-located with the BMG, or possibly an enhanced integrated BMG, which looks like an RNC to the UMTS core network (this is discussed further below).

A full mapping between BRAIN and UMTS protocols will be needed assuming that the MN to BAR interface is completely in line with BRAIN and that the interface between IWU and SGSN is completely in line with the UMTS specification.

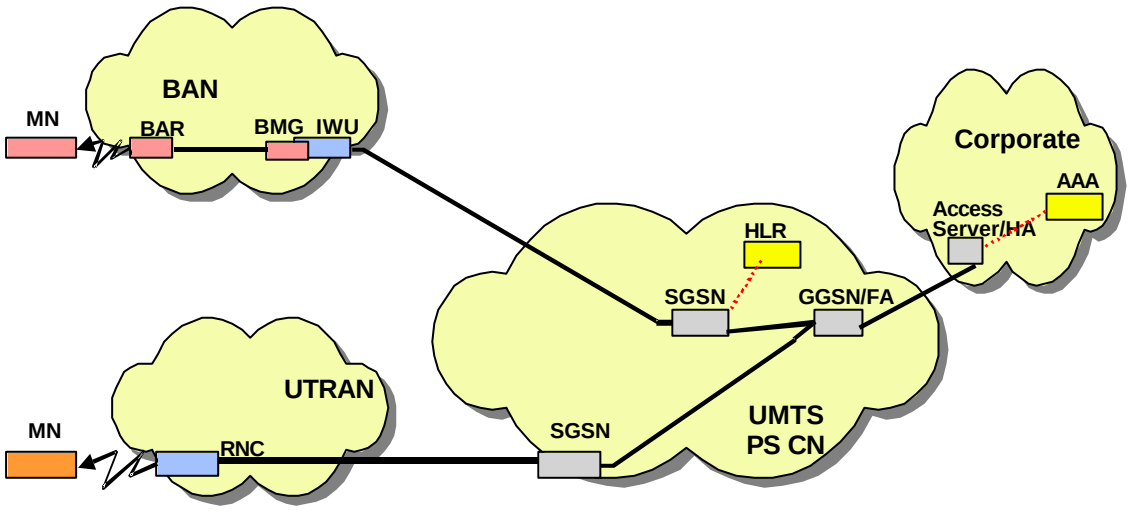


Figure A2-14: BRAIN Access to UMTS via the Iu interface (via SGSN)

Figure A2-15 shows the UMTS bearer concept with the HIPERLAN/2 access integrated. The UMTS Bearer is not changed in respect to the different radio interfaces. The Radio Access Bearer (RAB) must be adapted to the new, underlying Distribution Network (DN) Bearer which represents transport within the BAN, and the HIPERLAN/2 (H/2) Bearer. Both these bearers pass through the BAR, which is represented in the diagram as a HIPERLAN/2 Access Point.

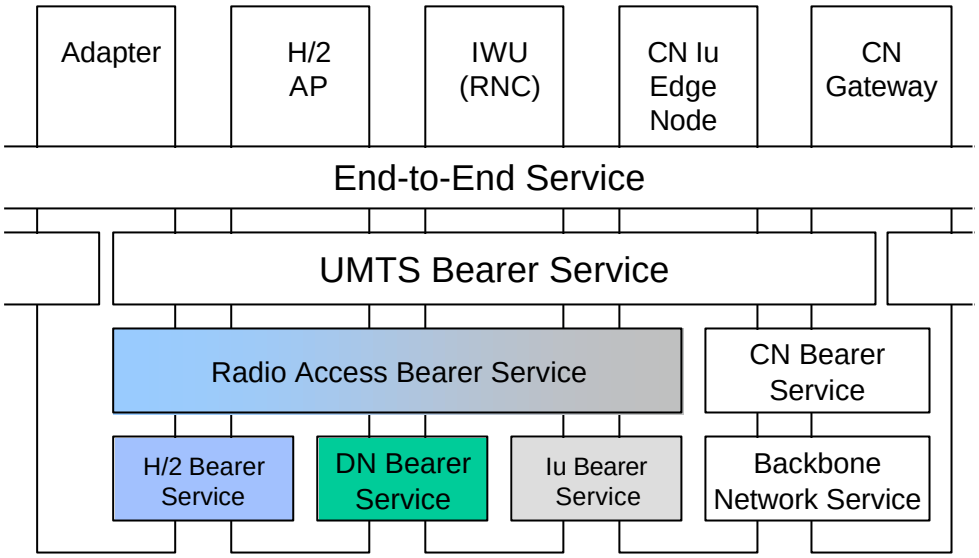


Figure A2-15: Adapted UMTS Bearer Concept

However, if a HIPERLAN/2 based radio access network is connected via the Iu interface, an Interworking Unit (IWU) is needed to exchange the packets between the BRAIN network and UMTS. The task of the IWU is similar to a Radio Network Controller (RNC) in UTRAN. It has to relay the Iu bearer service on the core network side to the distribution network bearer service on the other side. This includes an appropriate location and mobility management for the MNs in the HIPERLAN/2 coverage area. As depicted in Figure A2-16, the HIPERLAN/2 distribution network is provided by the BAN infrastructure, where the BARs located at HIPERLAN/2 access points (or access point controllers) operate at the IP level.

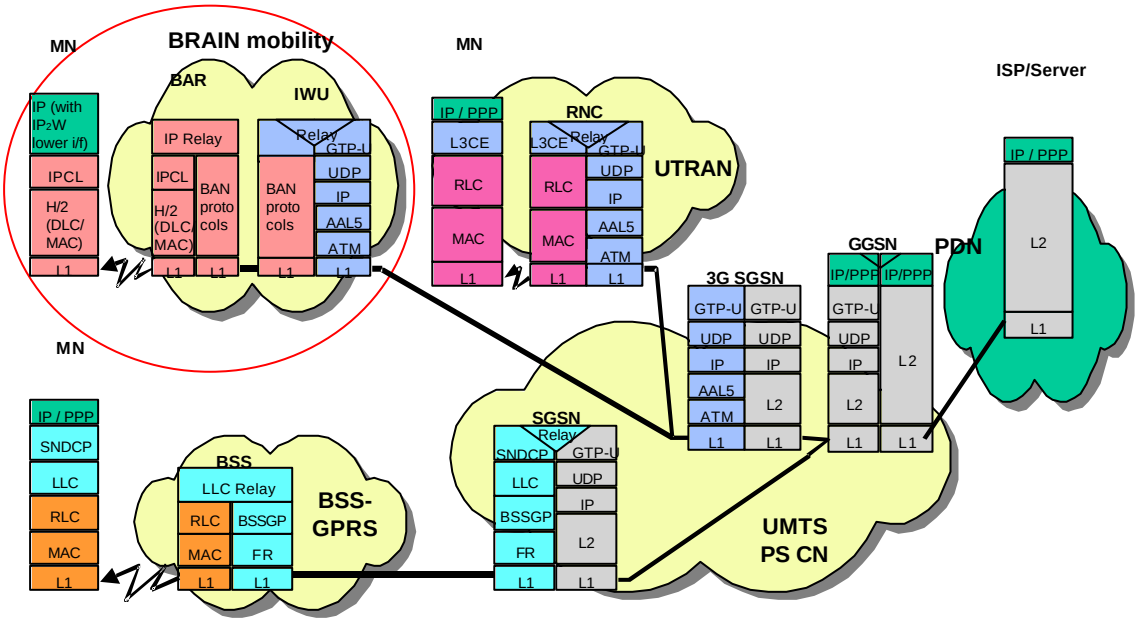


Figure A2-16: BRAIN – UMTS Protocol Stacks (user plane)

The Internet Protocol (IP) will be used to transfer packet switched data over the Iu interface as well as in the core network. The GPRS Tunnelling Protocol (GTP) on top of this transport IP layer provides a tunnelling service through the core network till the access network and encapsulates the user data, as also depicted in Figure A2-16. Hence, if IP datagrams are transmitted on user level, two IP layers exist in the packet-switched architecture within the core network.

With such tight coupling the speed of handover will be increased. This form of design is could be ideal for some operators as it allows for a core IP structure common to their networks and prevents duplication. This trend towards convergence would be then easier if further convergence with different technologies is

required. Again it will be essential that all parts of the core network can cope with the expected higher bit rates.

A2.3.1.4 Conclusions on Coupling

To aid clarity in the distinction between the different tight coupling options and the loose coupling option these can be seen in Figure A2-17. The Layer 2 mobility has to interoperate with layer 3 mobility in BAN in the tight coupling via the SGSN; this does not have to be the case in the tight coupling via the GGSN; and the BRAIN ISP represents the loose coupling.

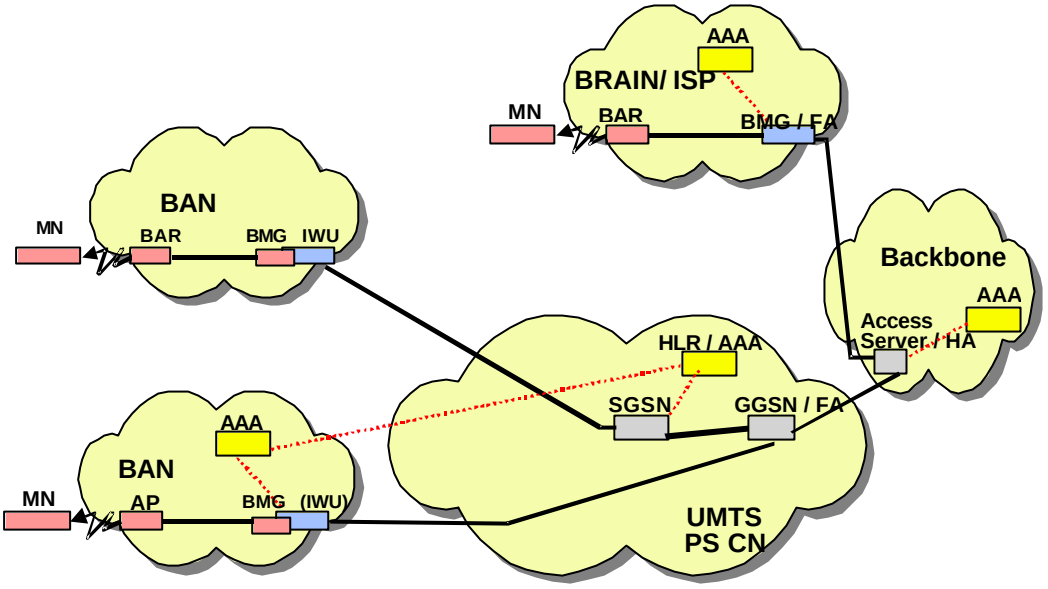


Figure A2-17: Combined View of all BRAIN Public Access Alternatives

For completeness we should mention that an alternative view of tight coupling would be to have the GGSN below the BMG in a network integration where the UMTS is seen as an alternative access technology on the BRAIN network. This is currently not being considered by ETSI but would be an alternative that might suit an operator without legacy UMTS infrastructure, or an operator whom has BRAIN dominant network. This is an unlikely approach to be adopted initially by an existing mobile operator.

We can concluded in the short report on the issue of coupling by looking in summary at the effect on some of the characteristics of the coupling schemes that have been detailed, see Table A2-1. These characteristics are now discussed in more detail in the following sections.

Characteristic/ Coupling	**No-Coupling**	**Loose Coupling**	**Tight Coupling (GGSN)**	**Tight Coupling (SGSN)**
Handover Speed	Slowest as longest path update route	Faster as shorter updates	Faster	Fastest
Security	Unknown	Good	Good	Good
Context Transfer	Slowest	Faster	Faster	Fastest
Operators	Ideal for independent	Trusted Partners	Single/Trusted Partners	This would really be for a single operator and is a complex issue with regards to its applicability in all cases,
QoS	Complete Renegotiation	Complete Renegotiation	Some Renegotiation	Minimal Renegotiation
Triggers for handover	Could come from a number of sources outside of the	Could come from external and internal	Will come from current network or MN	Will come from current network or MN

	current network.	sources		
IWU (Interworking Unit)	Not required	AAA–HLR interaction	AAA-HLR interaction	Full interaction between BRAIN and UMTS protocols

Table A2-1: Summary of Coupling vs. Characteristics.

A2.3.2 Other Issues/ Discussions

A number of issues related to the interoperation of BRAIN and UMTS networks have been considered and discussed during the work of this activity these will be looked at here. Some of these areas have had more consideration than others, also some have not reached a conclusion however the state of discussion will be included.

A2.3.2.1 Triggers

We consider what conditions would trigger a handover to or from a BRAIN network from or to a non-BRAIN network. These can be classified under four areas:

BRAIN Network Centric

Terminal Centric

Non-BRAIN Network Centric

CN Centric

Looking at each of these some of these have relevance only for a particular type of coupling this is thus indicated.

A2.3.2.1.1 BRAIN Network Centric

These are triggers that are started by the BRAIN network, this does not mean that the BRAIN network will control handover this may still be the MN it is just that the network has provided the trigger. These triggers will only occur whilst the mobile is on the BRAIN network. These triggers could occur in any of the coupling methods mentioned

?? Lack of bandwidth or resources demanded force handover

?? No(or Poor) service coverage

?? User profile related e.g. match to a profile attribute requesting that handled via UMTS connection

A2.3.2.1.2 Terminal Centric

These are triggers that the terminal might initiate. These triggers could occur whilst on the BRAIN or Non-BRAIN network. Again these triggers could occur in any of the coupling methods outlined, however in the case of tight-coupling the case of handover due to future demand would be judged on the case of the capabilities of the radio access rather than the network.

?? Better deal on other network for type of communications, probably set in the user profile of the terminal.

?? User Initiated handover will in turn be triggered from the terminal.

?? Detected radio interface need e.g. signal strength. Although this could be network triggered as the terminal will be capable of detecting multiple networks this choice could be terminal triggered.

?? Based on future demand e.g. about to send large file go to best network to sustain, again this is probably set in the user profile of the terminal.

A2.3.2.1.3 Non-BRAIN Network Centric

Triggers that are started by the Non-BRAIN network, this does not mean that the non-BRAIN network will control handover this may still be the MN it is just that the non-BRAIN network has provided the trigger.

?? Paging request received on other network for terminal. This would only occur in the no-coupling or loose coupling approaches and would assume that the terminal is simultaneously registered on both

networks. Then if in Idle mode on the other network the paging message would still be sent if any direct traffic to the terminal in the other network is received which could trigger handover.

?? Data waiting on other network and now in range. This again is specific to the no-coupling and loose coupling approaches, in the case where content is waiting on another network when the mobile moves in to range of the network and receives a paging signal.

A2.3.2.1.4 CN(Correspondent Node) Centric

Note: This could be the SIP proxy of the user outside of the BRAIN network. This could occur in any of the coupling methods however is most likely to occur in the no-coupling case.

?? Data matched to profile indicates that other access is better

?? Paging Request specific to other access

?? CN knows that its demands can only be meet on other interface

?? CN is on the other network and prefers same network connection

?? CN is admin on other network, authentication should be on the other network.

A2.3.2.1.5 Summary

These are just some of the triggers that may start or prompt for handover between a BRAIN and non-BRAIN network. As has been indicated some of these triggers will only apply in some of the coupling methods.. A tightly integrated approach will lead to a reduced number of triggers as parts of the networks such as paging and admin are tightly integrated. The adoption of a SIP solution in the no coupled solution could also be intelligent so as to reduce signalling E.g. directing traffic to the currently active network.

A2.3.2.2 Security

The issue of security in handover is important for any operator and subscriber; each of the coupling methods outlined has different security characteristics. In general it could be stated that the tighter the level of integration the greater the security retention is. In the no coupling case the security provisioning for each of the network is independent and hence there will need to be the creation of new security contexts between the MN and the network. In the case of the tightly coupled and loose coupling approaches the security contexts between the MN and the network may not need to be remade. The other key issue is if any security contexts between the MN and any CN can be retained, this is unlikely in the no-coupling approach. Unfortunately this is an issue that we have not been able to study in detail.

A2.3.2.3 Context Transfer

When referring to context transfer in this section we are only using the term Context Transfer in the generic sense, e.g. any context information (QoS etc) relating to the current session that needs to be transferred between the two networks. The BRAIN is already looking on the context transfer issue within the BAN (see sections on Mobility Management and QoS) however this work is not completed. To this extent it is to early to consider it in detail between 2 networks. The conclusions however is that any information that does need to be passed would have a longer path in some of the coupling method, hence the relative comments in Table A2-1. This issue of context transfer will be important in the future and should be considered as an outstanding issue for future work.

A2.3.2.4 Operators

The different forms of coupling are important it would be wrong to conclude that one method is the only way. The choice of coupling method is likely to be based on the background of the operator. Obviously it would be impossible to consider all the possible backgrounds of operators but they could be classed in to two rough categories, those who come from a legacy UMTS background and those whom are BRAIN oriented. There are further possible distinctions with the relative independence of the network operators.

With the no-coupling approach the two operators can be completely independent this is the only coupling option that allows this. As the coupling gets tighter the level of independency is reduced. In the tight coupling cases it is only really feasible that a single operator is running both networks.

Lets consider the legacy UMTS operators viewpoint they will have already invested in a UMTS based network and want to get the most value out of their existing infrastructure, this will be an evolutionary view rather than a revolutionary. To this end a tight integration would be ideal as it would enable to

operator to provide increase services to the user without a clear differentiation of aspects of service. It also allows for the operator to keep close control of their customers.

Lets consider the BRAIN oriented operator viewpoint this is someone whom is unlikely to have a legacy UMTS network, then the restrictions placed tight integration within a UMTS dominated network may not make sense. This is where either we will see a loose coupling or no-coupling approach (the traditional Internet view different service providers, freedom to build your own service portfolio). Other alternatives would be for a tight integration but with the UMTS being tied into the BRAIN network core structure, the alternative tight coupling talked about in section A9.1.4. This will require further studies.

A2.3.2.5 QoS

When vertical handover occurs between two different access technologies with very different capabilities it is unlikely that the same level of QoS will be experienced. Indeed this could be one of the drivers for the vertical handover in the first place. The issue with the QoS is how much of the QoS contexts would need to be renegotiated this is probably dependent on the amount of commonality between the old and new paths. Hence at high level more renegotiation is likely to be needed in the non-coupled approach. QoS issues in general are discussed in detail in the QoS section of the report.

A2.3.2.6 Dependency to UMTS releases.

The couplings shown here are assuming a R99 based UMTS network however standardisation already exists for parts of R4/R5 releases of UMTS these head towards an all-IP network which will look very different to that considered in this document. Indeed further releases post-R5 are likely to be IP based as far as possible. The handover issues discussed here have not been considered for such networks however with a native IP interface the type of coupling would look much more like the loose-coupling or no-coupling approach.

A2.3.2.7 Open issues for future studies.

We have, whilst considering the subject of interworking come across a lot of things that need and indeed it would be interesting to study further. These issues may be studied further within the MIND project. The areas where further consideration is necessary are:

?? The behaviour and complexity of tight coupling below the SGSN.

?? What are the alternatives for coupling with the BRAIN network being dominant, and what would this mean.

?? How will future UMTS releases affect the couplings outlined.

?? What is the user experience in the different coupling cases, e.g. the level of seamlessness.

?? Simulations to show the handover performance for the different cases.

?? To what extent can single mode BRAIN terminals connect to the BAN in the tight SGSN case.

?? What coupling cases would suit different types of operator, this has been started but is a complex task.

A2.3.3 BRAIN - UMTS Interoperation References

[A2.27] "Mobility Support using SIP", Elin Wedlund & Henning Schulzrinne; 2nd ACM/IEEE Int Conf on wireless and mobile multimedia (WoWMoM'99) Seattle, Washington, August 1999.

A2.4 Security

A2.4.1 Introduction

Terminal mobility induces several types of threats due to the ability of the mobile to connect to different foreign networks. Indeed, when it is away from home, the terminal (which is basically a computer with programmable capabilities) can behave as an attacker or as a victim of the other terminals connected to the access network. Moreover, a third party computer may exploit the fact that the terminal may be roaming to steal its identity and perform an attack on the home network.

One of the main tasks of the BAN is to provide security features enabling trusted use of the terminal. For instance, its quite obvious that the BAN will have to be able to operate AAA (Authentication, Authorisation, Accounting).

This section provides a discussion of security requirements for the BAN. First it lists the security objectives that can be expected by a BRAIN network operator, and also an end user. Then, it describes current AAA schemes defined in IETF and their integration within BRAIN (this is consistent with security constraints for Mobile IP given in [A2.28]). The next subsection describes the trust relationships that are required between different components of the architecture. The final subsection focuses on the security functions to be supported by the BAR (BRAIN Access Router).

A2.4.2 Security Objectives

This section details security issues specific to the support of the mobility on IP network. These are intended to apply to any IP (micro/macro)-mobility protocol.

A2.4.2.1 Confidentiality

All transferred data will be protected to the same level of confidentiality or higher as the home network. All mobility management information will be protected from any unauthorised disclosure, whether in transit or storage at any point in the network. The operators can protect packets they forward on behalf of their users by encrypting this traffic during exchanges with other operators, although true user data security is best assured independently using end-to-end techniques such as IPSec.

Special treatment should be applied for the air interfaces which are very prone to monitoring.

A2.4.2.2 Integrity

All transferred data and databases will be protected against any unauthorised modification or deletion. This is particularly important when exchanging information related to users' accounts and locations. It must also be possible to prove the integrity of data for non-repudiation purposes.

Consequently, the integrity of the following exchanged data should be checked:

?? AAA data between AAA infrastructures elements

?? Mobility data (for instance binding update for Mobile IPv6) between the mobile and its home network and its correspondents.

A2.4.2.3 Authentication

All network elements (nodes and clients) and network users will have associated authenticators that can be directly and unambiguously attributed to those elements and users. We assume that the home network would normally authenticate all information (packets) before allowing these to enter its internal network, in order to protect this internal network and its users, although this cannot be enforced by the BAN itself.

The BAN should be able to authenticate the MN or the user itself before the BAR provides IP connectivity to the MN.

A2.4.2.4 Authenticity

The operators must make sure that all its stored information is authentic. The operators must be able to verify the authenticity of users requiring their services. This does not necessarily mean that the operators must authenticate the user directly (which would mean that the identity of the user is revealed to the operator in question).

A2.4.2.5 Availability

All information, resources and network services will be protected against DoS (Denial of Service) attacks.

The operator must provide services to its users according to a given policy. In general users should have rights giving them a guaranteed access to the services which they have paid for.

Moreover, a malicious MN (or a MN which has failed to authenticate) should not be able to flood the mobility management equipment of the BAN with malicious packets (for instance, attempt to create large numbers of sessions).

A2.4.2.6 Authorised Access

An operator must ensure that users are authorised to access the services that they require.

Users that have subscribed to services on their home network should be able to access these services at the visited operator (for example in case of roaming). This requires that the operator is able to get some kind of authorisation from the visitor's home network, or another trusted party.

All operators must ensure that all claims to access confidential data are requested by those authorised to do so **before** granting access to this **data**.

A2.4.2.7 Accountability

All network user actions will be directly and unambiguously attributable to those users responsible. The operator must make sure the users can be held accountable for their use of resources (see AAA subsection below).

A2.4.2.8 Location privacy

Correspondent hosts might be able to get information on the location of the MN. For instance, the care of address includes the BAN (or more generally foreign network in the case of MIP) prefix. If a correspondent is able to map the prefix to a physical network (for example, a specific BAR), it could also deduce the geographical location of the MN. A MN must be able to hide its (current or past) location from any correspondents.

In addition, signalling over the air interface which includes the identity of a user (e.g. implicitly by reference to a MN's home address) should be encrypted to prevent air interface monitoring being used to identify users. An alternative is to ensure such signalling (e.g. in paging messages) uses temporary identifiers only, and not a user's permanent identity.

A2.4.3 AAA

Authentication, Authorised access and Accountability from the previous list are known as AAA. The Internet Engineering Task Force has defined the global architecture of the AAA for Mobile IP. At the moment, the most probable IETF candidates for the AAA protocol are DIAMETER [A2.29], RADIUS [A2.30] or COPS [A2.31].

A2.4.3.1 Basic AAA Model

Within the Internet, a client belonging to one administrative domain, the home domain, often needs to use resources provided by another administrative domain, the foreign network.

An agent in a foreign domain, the attendant, being called on to provide access to a resource by a mobile user, is likely to request or require the client to provide credentials which can be authenticated before accesses to resources are permitted. The attendant often does not have direct access to the data needed to complete the transaction. Instead, the attendant is expected to consult an authority (typically in the same foreign domain) in order to request proof that the client has acceptable credentials. Since the attendant and the local authority are part of the same administrative domain, they are expected to have security relationships that enable them to securely transact information locally, or are minimally assumed to be able to dynamically establish a security association which will persist for the required lifetime.

The local authority (AAAL) itself may not have enough information stored locally to carry out the verification for the credentials of the client. In contrast to the attendant, however, the AAAL is expected to be configured with enough information to negotiate the verification of client credentials with external authorities. The local and the external authorities should be configured with sufficient security relationships and access controls so that they, possibly without the need for any other AAA agents, can

negotiate the authorisation that may enable the client to have access to any/all request resources. In many typical cases, the authorisation depends only upon secure authentication of the client's credentials. The basic relationship between client, attendant, and AAA agents is shown in Figure A2-18.

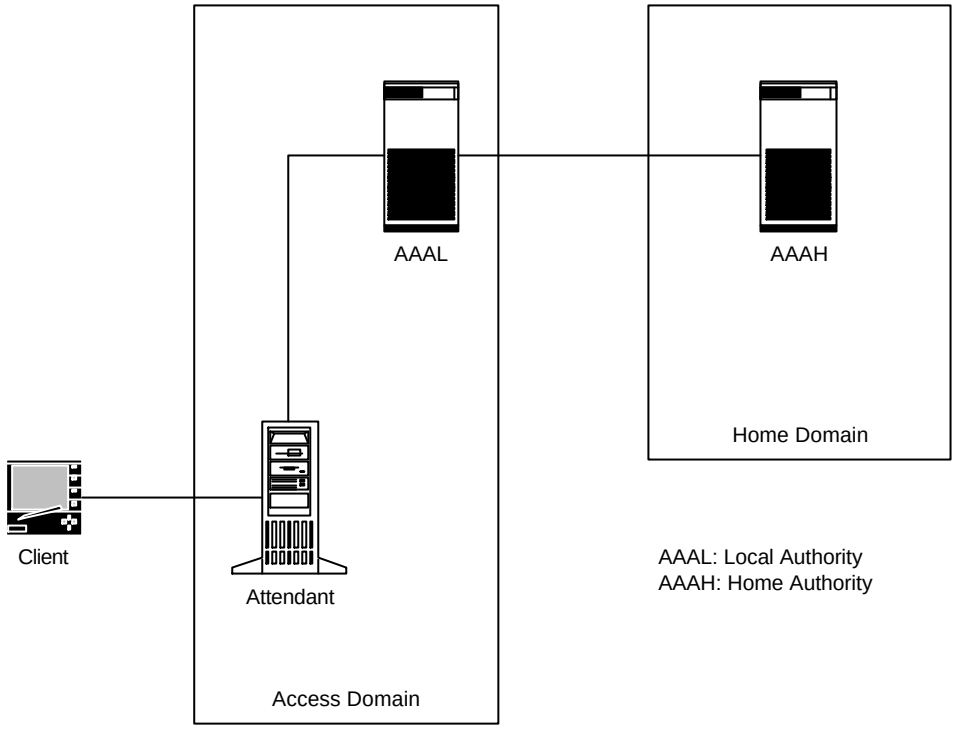


Figure A2-18: Basic AAA Components

Once the authorisation has been obtained by the local authority and the authority has notified the attendant about the successful negotiation, the attendant can provide the request resources to the client. Credentials allowing authorisation at one attendant should be unusable in any future negotiations at the same or any other attendant.

There might be many attends for each AAAL, and there might be many clients from many different Home Domains. Each Home Domain provides an AAAH that can check credentials originating from clients administered by that Home Domain.

As an example in today Internet, we can cite the deployment of RADIUS (Remote Authentication Dial In User Service) to allow mobile computer clients to have access to the Internet by way of a local ISP. The ISP wants to make sure that the mobile client can pay for the connection. Once the client has provided credentials (e.g. identification, unique data and an secure signature), the ISP checks with the client's home authority to verify the signature and to obtain assurance that the client will pay for the connection. Here, the attendant function can be carried out by the NAS, and the local and home authorities can be RADIUS servers, with all communication between attendant and AAA agent done using the RADIUS protocol.

From the description we can identify several requirements:

?? Each local attendant has to have a security relationship with the local AAA server (AAAL).

?? The local authority has to share, or dynamically establish, security relationships with external authorities that are able to check the client credentials (between AAAL and several AAAHs).

?? The attendant has to keep state for pending client request while the local authority contacts the appropriate external authority[15].

?? Since the mobile node may not necessarily initiate network connectivity from within its home domain, it must be able to provide complete, yet secure credentials without ever having been in touch with its home domain.

[15] The amount of state to be kept depends on the protocol used. However, even relatively stateless protocols (such as RADIUS) still require state to be kept at the attendant end.

?? Since the mobile node's credentials have to remain secure, intervening nodes (e.g. including the attendant, the local authority (AAAL) and any other intermediate nodes) must not be able to store any information which may enable them to reconstruct and reuse the credentials.

From this last requirement we can see the reasons for the natural requirement that the client has to share, or dynamically establish, a security relationship with the external authority in the Home Domain.

In addition to the requirements listed above, we specify the following requirements which derive from operational experience with today's roaming protocols:

?? There are scenarios in which an attendant will have to manage requests for many clients at the same time.

?? The attendant must protect against replay attacks.

?? The attendant equipment should be as inexpensive as possible, since it will be replicated as many times as possible to handle as many clients as possible in the foreign domain.

?? Attendants should be configured to obtain authorisation from a trusted local AAA server (AAAL) for Quality of Service requirements requested by the client.

In this section we will detail additional requirements based on issues discovered through operational experience of existing roaming RADIUS networks. The AAA protocol MUST satisfy these requirements in order for providers to offer a robust service. (These requirements have been identified by TR45.6 as part of their involvement with the Mobile IP working group.)

?? Support a reliable AAA transport mechanism.

- There must be an effective hop-by-hop retransmission and failover mechanism so that reliability does not solely depend on end-to-end retransmission
- This transport mechanism will be able indicate to an AAA application that a message was delivered to the next peer AAA application or that a time out occurred.
- Retransmission is controlled by the reliable AAA transport mechanism, and not by lower layer protocols such as TCP.
- Even if the AAA message is to be forwarded, or the message's options or semantics do not conform with the AAA protocol, the transport mechanism will acknowledge that the peer received the AAA message.
- Acknowledgements should be allowed to be piggybacked in AAA messages
- AAA responses have to be delivered in a timely fashion to prevent timeouts and retransmissions.

?? Transport a digital certificate in an AAA message, in order to minimise the number of round trips associated with AAA transactions The certificates could be used by foreign and home agents to establish an IPSec security association to secure the mobile node's tunnelled data. In this case, the AAA infrastructure could assist by obtaining the revocation status of such a certificate (either by performing online checks or otherwise validating the certificate) so that home and foreign agents could avoid a costly online certificate status check

?? Provide message integrity and identity authentication on a hop-by-hop (AAA node) basis

?? Support replay protection and optional non-repudiation capabilities for all authorisation and accounting messages.

?? Support accounting via both bilateral arrangements and via broker AAA servers providing accounting dearinghouse and reconciliation between serving and home networks. There is an explicit agreement that if the private network or home ISP authenticates the mobile station requesting service, then the private network or home ISP network also agrees to reconcile charges with the home service provider or broker. Real time accounting must be supported. Timestamps must be included in all accounting packets.

We place the following additional requirements on the AAA services in order to satisfy the clients that programmed to receive some IP-specific resources during the initialisation phase of their attempt to connect to the Internet.

?? Either AAA server must be able to obtain, or to co-ordinate the allocation of, a suitable IP address for the customer, upon request by the customer.

?? AAA servers must be able to identify the client by some means other than its IP address.

A2.4.3.2 Basic BRAIN Architecture

As the BRAIN architecture and mobility protocols are not completed, we assume for this document that the basic mobility architectures relies on Mobility Agents such as it is done in Mobile IP as well as in current micro-mobility drafts (Cellular IP, HAWAII, HMIPv6…). Thus, we consider that a Mobility Agent is located in the Home Domain and aware of the current location of the MN. This Mobility Agent is a pure functional entity which may be included in a server which is not Mobile IP compliant, if later the BRAIN macro-mobility is not based on Mobile IP protocol. We consider also that terminal mobility is handled inside the BAN and specifically that the BAR which is the first router seen by a MN implements key features supporting mobility within the BAN.

Then, different mobility schemes can be mapped on this generic approach, such as for example:

?? Basic Mobile IPv4 and v6.

?? Mobile IPv6 and extensions such as HMIP

?? Mobile IPv4 + use of CCOA where the FA functions are quite limited

?? HAWAII which is closed to MobileIPv4 + CCOA and introduces per host forwarding in the local domain.

?? Cellular IP which is designed to be used with Mobile IP for macro-mobility handling.

?? Other micro/macro mobility schemes not specifically based on Mobile IP.

MNs require specific features from the AAA services, in addition to the requirements already mentioned in connection with the basic AAA functionality and what is needed for IP connectivity. For instance the AAA services must provide

?? Cryptographic material for air interface ciphering,

?? Authentication for home registration.

For application to BRAIN, we modify the general model. The main adaptation is that the Attendant functions are handled by the BAR.

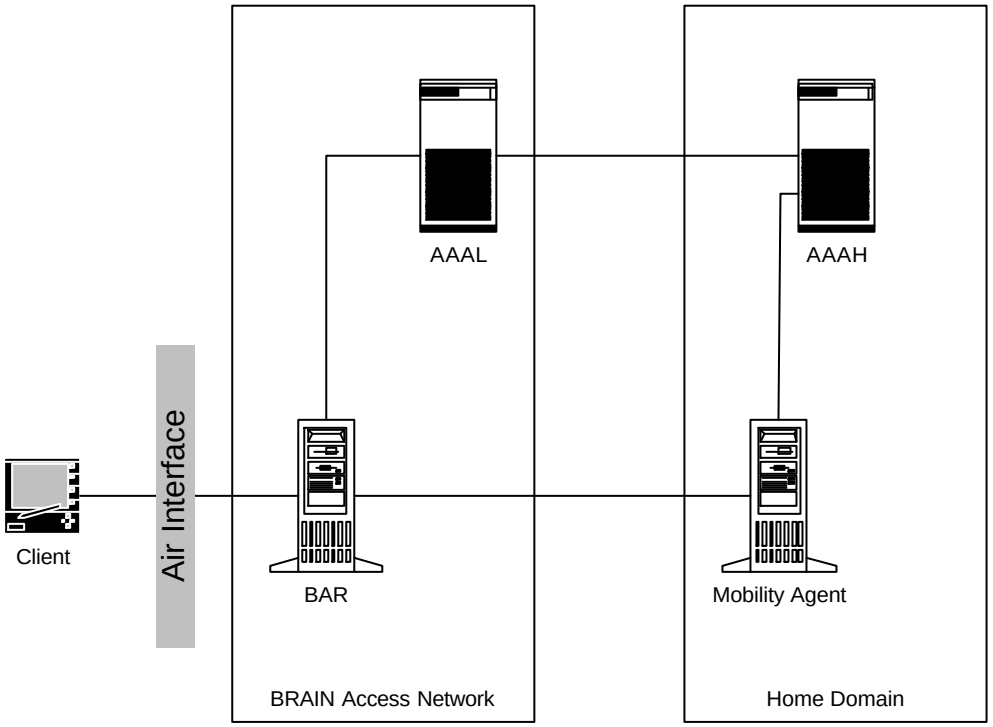


Figure A2-19: AAA Model Adapted to BRAIN

In the AAA Basic model the initial AAA transactions are handled without needing the home agent. However, in BRAIN, we must accept that the access request also has to be processed by the remote AAAH server, possibly in conjunction with the Mobility Agent. These are therefore implicitly involved in the AAA procedures along with the BAR, although the BAR does not have direct visibility of this.

This means that during the initial access, something has to happen that enables the Mobility Agent and the BAR to perform subsequent access and registration, but this is just the set-up and authorisation of IP traffic from the MN. After the initial registration, the AAAH and AAAL would not be needed, and subsequent registrations would only follow the lower control path between the BAR and the Mobility Agent; these are seen by the BAR as pure IP flows.

In particular, note that this allows macro mobility protocols to continue to evolve rapidly in the Internet, without impact on the security architecture for devices at the level of the BAR (of which there could be hundreds of thousands in the world). This is why it is convenient also to have all the 'higher level' security functions running between the AAAL and other servers, which are mainly transparent to the BAR.

We propose that if it is required to some kind of optimised combined access and mobility registration procedure, that this should be implemented as additional functions within the AAAL. This allows the function to configured centrally and made dependent on subscriber database information. It should be possible to implement such functionality in a relatively generic and 'future proof' fashion (e.g. "when receiving an access request from user X from domain Y, invoke procedure Z [which might be some kind of registration procedure]"). The functions of this type within the AAAL should be able to exploit any standard current or future protocols from fixed IP networks.

After the initial registration, the mobile node is authorised to continue using the local mobility protocol implemented in the BAN without requiring further involvement by the AAA Servers. Thus, the initial registration will probably take longer than subsequent registration. However, note that while the MN remains attached to the BAN it is able to retain the same IP address (according to the general BRAIN architecture) so it may be that no further registrations are needed anyway. Registrations may only be needed for soft-state refresh in the macro mobility protocol (but then these are not time critical).

In order to reduce this extra time overhead as much as possible, it is important to reduce the time taken for communications between the AAA servers. A major component of this is the delay induced by the time taken to cross the wide-area Internet that is likely to separate the AAAL and the AAAH. This leads to a further strong motivation for integration of the AAA functions themselves, as well as integration of AAA functions with the initial registration. In order to reduce the number of messages that cross the network for initial registration of a MN, the AAA functions in the BAN (AAAL) and the home network (AAAH) need to interface with the BAR and the home agent to handle the registration message. Latency would be reduced as a result of initial registration being handled in conjunction with AAA and the mobility agents. Another way to reduce latency as to accounting would be the exchange of small records.

The AAA home domain and the 'mobility' home domain of the MN need not be part of the same administrative domain. Such a situation can occur if the home address of the MN is provided by one domain, e.g. an ISP that the mobile user uses while at home, and the authorisation and accounting by another domain, e.g. a credit card company. The BAR sends only the authentication information of the mobile node to the AAAL, which interfaces to the AAAH. After a successful authorisation of the mobile node, the BAR is able to continue with the mobile registration procedure. Such a scheme introduces more delay if the access to the AAA functionality and the mobility protocol is serialised. However this situation might be avoided in the BRAIN model.

All needed AAA and mobility registration functions should be processed during a single Internet crossing. This must be done without requiring AAA servers to process protocol messages sent to HA and BAR. The AAA servers must identify the BAR and the HA and security associations necessary to process the mobility registration, pass the necessary registration data to those Mobility Agents, and remain not involved in the routing and authentication processing steps particular to mobility registration.

For Brain, the AAAL and the AAAH servers have the following additional general tasks:

?? Enable authentication for mobility registration (TBD see above)

?? Authorise the MN to use at least the set of resources for minimal mobility and connectivity functionalities, plus potentially other services requested by the MN.

For instance, the MN must be able to

?? have access to the air interface and understand informational broadcast from the BAR (e.g. Router Advertisement, Challenge data for authentication, etc…)

?? should be able to talk to the BAR in order to initialise the AAA authentication.

But before registration and authentication are performed, the MN must not be able to

?? communicate with other node than the BAR (e.g. other MN, or equipment "behind" the BAR)

?? Initiate accounting for service utilisation

?? Use AAA protocol extensions specifically for including mobility registration messages as part of the initial registration sequence to be handled by the AAA servers (TBD see above).

In order to enable subsequent registrations, the AAA servers must be able to perform some key distribution during the initial registration process from any particular administrative domain.

A2.4.3.3 BRAIN with Local Home Agents

It is also possible the HA be located in the BAN. As long as the MN can get an address routable within the current BAN (be it publicly, or privately addressed) it can use the local mobility protocol to roam inside that domain, calling the BAN on which it booted (or attached) its temporary home. This address is likely to be dynamically allocated upon request by the MN.

In such situations, when the client is willing to use a dynamically allocated IP address and does not have any preference for the location of the home network (either geographical or topological), the local AAA server (AAAL) may be able to offer this additional allocation service to the client. Then, the home agent will be located in the local domain, which is likely to be offer smaller delays for new Mobile IP registrations. Apart from the changes in the AAAL, this scenario has no impact on the remainder of the BAN security architecture.

A2.4.3.4 Broker Model

In the previous configurations, the local and the home authority have to share trust. Depending on the security model used, this configuration can cause a quadratic growth in the number of trust relationships, as the number of AAA authorities (AAAL and AAAH) increases.

A broker may play the role of a proxy between two administrative domains which have security associations with the broker, and relay AAA messages back and forth securely. Alternatively, a broker may also enable two domains with which it has associations, but which do not themselves have a direct association, in establishing a security association, thereby bypassing the broker for carrying the messages between the domains. This may be established by virtue of having the broker relay a shared secret key to both the domains that are trying to establish secure communication and then have the domains use the keys supplied by the broker in setting up a security association. However, the redirection broker will usually require a copy of authorisation messages from the home domain and accounting messages from the serving domain, in order for the broker to determine if it is willing to accept responsibility for the services being authorised and utilised. If the broker does not accept such responsibility for any reason, then it must be able to terminate service to a MN in the serving network. In the event that multiple brokers are involved, in most situations all brokers must be so copied. This may represent an additional burden. Though this mechanism may reduce latency in the transit of messages between the domains after the broker has completed its involvement, there may be many more messages involved as a result of additional copies of authorisation and accounting messages to the brokers involved. There may also be additional latency for initial access to the network, especially when a new security association needs to be created between AAAL and AAAH (for example, from the use of ISAKMP). These delays may become important factors for latency- critical applications.

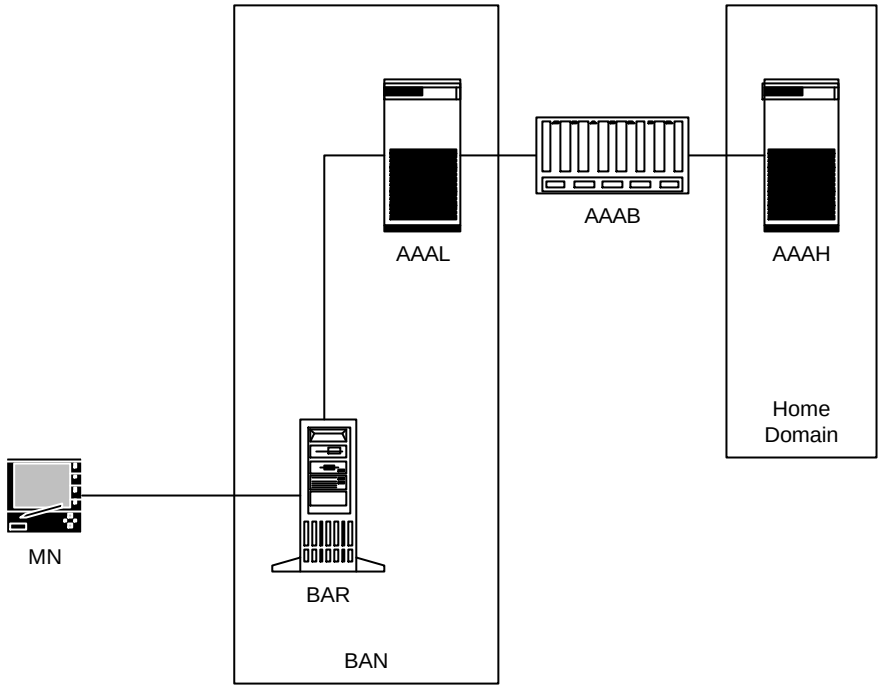


Figure A2-20: AAA Brokers

The following requirements come mostly from brokers in the particular case of authorisation for roaming dial-up users.

?? allowing management of trust with external domains by way of brokered AAA.

?? accounting reliability. Accounting data that traverses the Internet may suffer substantial packet loss. Since accounting packets may traverse one or more intermediate authorisation points (e.g., brokers), retransmission is needed from intermediate points to avoid long end-to-end delays.

?? End-to-end security. The Local Domain and Home Domain must be able to verify signatures within the message, even though the message is passed through an intermediate authority server.

?? Since the AAAH in the home domain MAY be sending sensitive information, such as registration keys, the broker MUST be able to pass encrypted data between the AAA servers.

A2.4.3.5 Fast Handover

Since the movement from coverage area to coverage area may be frequent in BRAIN networks, it is imperative that the latency involved in the handoff process be minimised. After initial authorisation in a domain, further authorisations should be done locally within the local domain. When a MN moves into a new foreign subnet as a result of a handover and is now served by a different BAN, the AAAL in this domain may contact the AAAL in the previous domain to verify the authenticity of the MN. Alternatively, there is nothing to stop a single AAAL being shared across different BANs of the same administration.

A2.4.4 Trust relationship

A2.4.4.1 Dynamically established SA

The MN and the different BRAIN equipments must share trust relations. For instance, these relations can be instantiated as security associations (SA) of the IPSec model. Thus they are not only shared secrets but also data structures indicating the cryptographic algorithms, the algorithms parameter, the key life time of the relation.

Because of the huge number of MNs and BARs it is not realistic to consider a preconfigured SA between each MN and BAR. Consequently, these entities should be able to automatically establish SA. The IETF Internet Key Exchange protocol (IKE) can provide such a service. However, the BRAIN entities have to be able to prove their identities to each other through a certificate. A consequence on the radio link is that it is not possible to cipher the data at the radio level all the time. The radio link will be ciphered only once the SA has been set up.

Alternatively, we could consider that the dynamic SA runs end-to-end between the MN and AAAL, which eliminates the need to re-establish the SA on handover. In this alternative, the AAAL must be able to remotely configure the BAR so as to allow/disallow access from a given MN.

Note that here we are relying on air interface encryption only weakly in this case; the stronger security requirements for AAA (in particular, mutual authentication) require the type of security services provided by IPSec. Air interface encryption is used if at all only to enhance things such as location privacy.

Considering the AAA model, any pair of AAAL and AAAH must be able to dynamically set up an SA. If a broker is used, the AAA entities must share a static SA with the broker and these SAs will be use to build the direct SA between them.

A2.4.4.2 Reconfigured SA

Some of the SAs of the BRAIN Architecture do not need to be dynamically established. Dynamic establishment of SAs should be avoided as often as possible because of the delay induced by this kind of exchange. For instance, this is the case of the SA between the MN and its HA.

The BAR and the AAAL, if located on the same BAN, can also share a preconfigured SA as well as the AAAH and the HA.

A2.4.4.3 Global relationship

Figure A2-21 gives the trust relations between the BRAIN security entities

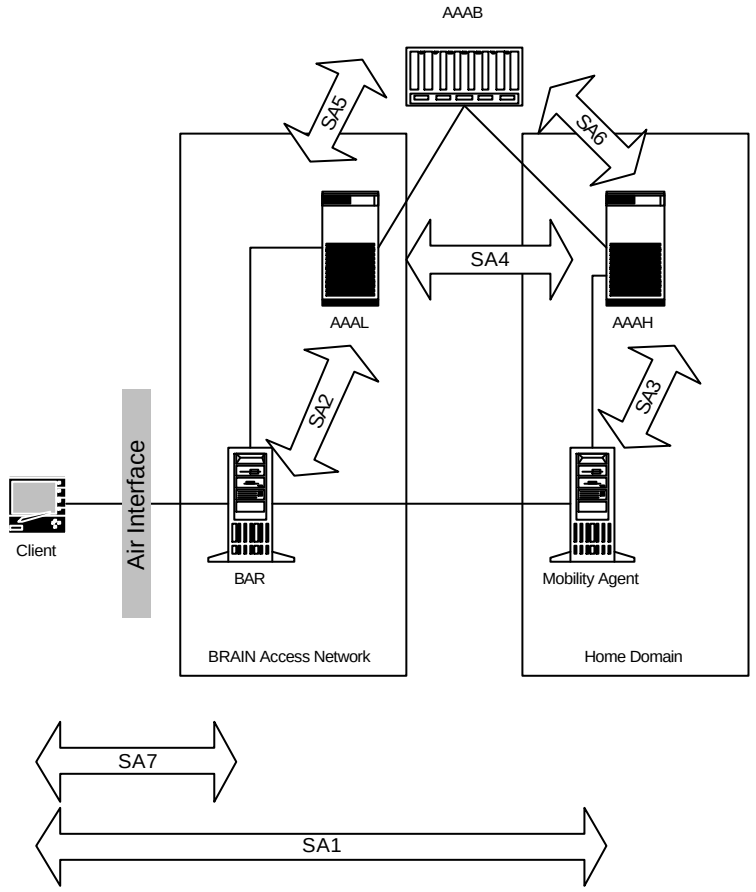


Figure A2-21: Security Associations

SA1 is the most obvious trust relation. However, some mobility protocols like Mobile IPv6 include a mechanism for a mobile to dynamically learn the address of its home agent. In this situation, SA1 is no longer static but must be dynamic.

The Broker is not required if the AAAL and the AAAH know each other. In this case, SA5 and SA6 are not needed because SA4 is already set up. If the broker is required, the SA4 set up is based on the SA5 and SA6.

SA7 is dynamically established after the BAR has checked the identity of the MN with the AAAL (which forwards the authentication to the AAAH). For optimisation reason, the mobility registration should be performed simultaneously with the SA7 establishment. SA7 will also be used by the BAR (or the radio access points) and the MN to cipher the data transmitted on the radio link. It must be pointed out that the establishment of SA7 also enables the MN to authenticate the BAR. This authentication avoid some kinds of attacks based on the introduction of malicious BS aiming at stealing the packets sent by the MN to the BAN.

A2.4.5 BAR Security Functions

The BAR plays a major role in the security of the mobility. The BAR is the first equipment in "direct view" with the UE. Consequently, during the connection phase, it has to be involved in all the authentication, registration, configuration services for the BAN and the MN. This means either that the BAR itself controls these procedures, or that the procedures take place with an other entity (like the AAAL), which is then able to set security controls in the BAR, as discussed above.

A2.4.5.1 Protecting the BAN

The BAR is the only screen in front of the BAN that can protect it. Intruders willing to get access to the BAN through the air interface must be stopped by the BAR.

When a MN connects to the radio access points the only connection the MN is allowed to perform is with the air interface of the BAR. Any connections with a machine beyond the BAN side interface of the BAR must be prohibited. The default configuration of the BAR must be to block any packet from the air interface to the BAN interface. If the MN authentication succeeds, a routing and an ARP (or neighbour discovery) cache entries must be open in the BAR. However it is possible that the authentication succeeds but the MN does not have rights to have access to the full Internet. In this case the routing table of the BAR should be configured in accordance with those rights.

A2.4.5.2 Challenging

In order to allow the MN to present credential to the AAA, the BAR must send challenge to the MN. The MN must respond with the signed challenge. It signs the challenge with its private key and adds its identity (and possibly its certificate). Depending on radio bandwidth management, the challenge can be regularly broadcast on the radio link or be sent to the MN on request.

A2.4.5.3 Forwarding credential and mobility registration

The BAR must forward the credentials to the AAAL. In the generic AAA model the BAR plays the role of the attendant.

For optimisation reasons, it is preferable that the mobility registration be send in the same packet as the authentication material. If this optimisation is enabled, the registration must be sent to the AAAL with the credentials. (TBD see above.)

A2.4.5.4 Provide confidentiality on the air interface

Once this authentication/registering procedure is computed by the AAA/HA, the BAR and the MN should establish an SA, for instance with IKE.

This SA should be use to authenticate packet from and to the MN but also to provide confidentiality on the radio link (i.e. ciphering). This ciphering may be performed at IP level or radio level.

A2.4.6 Encryption on the radio link

A2.4.6.1 Available security devices

Layer 3 security (IPSec):

The IETF requires that any IPv6 node must also implement IPSec. Considering other Brain working group recommendations, it seems now that IPv6 will be mandatory in Brain. Consequently we can assume that IPSec is available on every Brain node. Moreover, it is reasonable to assume that IPSec can based its authentication and session key exchange on asymmetric keys.

Layer 2 security:

In parallel, the L2 may have its own security module. For instance a HIPERLAN/2 node is able to authenticate and encrypt. Here we should assume that L2 authentication and encryption is based on a

secret key (This is the weaker assumption, so it is the most reasonable. It has consequences on roaming. Remark: in GSM, there is only one secret key that can be used for authentication and to derive a session key for encryption.)

As a result, the data can be encrypted on the radio link at two different layers: L2 or L3.

However, it seems inefficient to encrypt twice (once for each layer). So if it is possible, one of the layers should be preferred.

A2.4.6.2 Which security device should be preferred?

As far as we know, no radio network standards require data ciphering and authentication over the air with IPSec, whereas a L2 radio access could require the radio link to be encrypted, for instance if GSM security is used at L2. Consequently, the following table sums up the different situations that should be taken into account for the security analysis. Global security refers to the security rules resulting from the conjunction of the security policy of the Mobile and the one of the BAN. Of course these policies must be sound and consistent, otherwise the security cannot be applied (e.g., a MN requires encryption while the AP forbids it).

	Global security requires encryption of the radio link	Global security does not requires encryption of the radio link
L2 encryption is mandatory	Use L2 encryption	Use L2 encryption
L2 encryption module is available but encryption is optional	Use L2 encryption (*)	Do nothing for air encryption
L2 does not implement any encryption	Use IPSec	Do nothing for air encryption

Table A2-2: Security rules

(*) This choice is almost arbitrary. However, if L2 can do encryption without authentication (e.g. : with a Diffie-Hellman algorithm), L2 encryption is more efficient.

A2.4.6.3 Defining a global security policy

In order to express their security needs the MN and the BAN nodes should embed a Security Policy Database (SPD). These databases are a list of security rules targeting L2 or L3 security. This Brain SPD could be defined by the constructor, the operator, and could be completed by user applications or by the user himself.

For instance, the APs could mandate that the mobile nodes must connect through them with L2 encryption. Considering that the different Brain operators could have different level of security requirement, it cannot be assume than the SPD on the BAN depend only on the radio technology.

A2.4.6.4 How can IPSec be used?

1) The user is able to cipher all its connections with all its correspondents. In this case end-to-end encryption also guarantees radio ciphering.

2) One of the correspondents does not want to cipher. In this case Brain must provide the MN with a tunnel end point located in the BAN. Which node should play this is not clear today and depends on the Mobility Management protocol implemented on the BAN.

A2.4.6.5 Impact of IPSec ciphering on hand-over

In the case the MN has an end-to-end encrypted tunnel with all its correspondents, nothing special has to be done while hand-offs. In the case the MN uses a Brain-Tunnel end point, it may happen that the end-point has to be moved. This question cannot be solved here, it depends on the MM protocol.

A2.4.6.6 Impact of L2 ciphering on hand-over

When a MN hand-offs from a AP to an other AP, the new AP has to get the secret key that will enable it to continue the encryption/decryption of the MN data. This can be achieved in two ways: the new AP

retrieves the key from the key database or the old AP transfers it to the new AP. The first solution is time consuming in the case the key database is not located on the BAN (roaming). The second solution needs an efficient context transfer protocol.

A2.4.6.7 Impact of L2 ciphering on the IP2W interface

?? The L2 must give information (e.g.: MN identity, L2 encryption enable, L2 encryption mandatory, etc…) to L3 and to the security policy module management. The AP and the MN security module must behave according to 1.2: start L2 key retrieval, start IPSec, skip encryption.

?? For roaming as for hand-over, APs have to get the secret key from an other node of the BAN (old AP or database). This key can be transferred securely over the IP layer using an IPSec tunnel for instance. But the AP stack must be able for downward the key from L3 to L2.

Remark: The L2 authentication management can be done similarly to L2 encryption.

Remark: It is up to IPSec to deal with VPN and end-to-end encryption. The L2 secret key should not be used for another purpose than for authentication and encryption on the air.

A2.4.7 Authentication between the MN, the BAN and the home network

A2.4.7.1 Authentication at L3

L2 authentication is not sufficient: the MN authenticates the Network but the authentication is not mutual. The authentication of the BAN to the MN is not possible at L2 because the MN cannot retrieve the secret key of the BAN if it is roaming (it would need a connection to get this key and it needs to authenticate to set up the connection). Mutual authentication can ensure the mobile it does not connect to a false AP.

As a result, the L2 link must be established first (this may require L2 authentication and L2 encryption) and then the mobile must authenticate to the BAN, the Home Network and finally perform the home registration. In parallel, the BAN can authenticate to the MN.

A2.4.7.2 What to do if authentication fails?

If one of the authentications of the MN to the BAN fails, the BAN must be able to disconnect the MN or at least limiting the use of the resources by the MN. For instance, with current ISP connections, if the authentication fails, the modem of the ISP hangs up. Brain networks should be able to do something similar:

?? If the AP can brake the L2 connection, then it must do it.

?? If the AP can't brake the L2 connection (e.g. IEEE802.11), then the closest Brain Router to the MN should stop all the IP connections of the MN. This way, the MN can't use the BAN. However this would not prevent the MN from doing some attacks like Denial of Service on the radio interface or intrusion on other MNs connected to the same access point. According to the Brain architecture reference model, this role is play by the BAR. In the AAA terminology, this entity is called the attendant (see D2.1 for a global description of AAA for mobile networks)

A2.4.7.3 How to do the authentication?

A2.4.7.3.1 MN - BAN

To authenticate efficiently, a node A sends a random challenge to a node B. B signs this challenge with its own key and send the result to A, then A checks the signature:

?? In the secret key case, A signs the challenge and compares it with the result transmitted by B. If they are equal the authentication succeeds

?? In the public key case, A applies the public key of B to the result it has received and checks that it gets the challenge.

This technique gives the replay protection and should be used for the mutual authentication between the BAN and the MN. If the challenge can be send trough IP2W immediately after the L2 authentication/encryption took place (thanks to a special primitive to be defined) it should be done. Otherwise the challenge should be broadcast by the BAN over the air regularly or request by the MN.

The MN can send its challenge to the BAN (piggybacked) during its own authentication phase.

A2.4.7.3.2 MN – HA

The authentication with the HA is done by checking the-AH of the binding update and binding acknowledge messages.

A2.4.8 How to get the material to sign/check: keys, certificates, etc…?

The MN has now way to get extra material when it connects: the link is not established so it can't use it. We focus on the authentication of the mobile to the BAN.

A2.4.8.1 At L3

The BAR should delegate the key retrieval to a dedicated node: the AAAL. This node can check the validity of the MN authentication and controls the attendant (the BAR). It orders it to authorise or refuse the connection. If the AAAL doesn't have the necessary information on the MN, it can consult a server on the home network of the mobile (AAAH). This is useful if the MN is roaming or if the BAN does not match a Brain administrative network.

A2.4.8.2 At L2

Today, it is not clear if the AP can also delegate the L2 authentication to the AAA architecture as well. However, the trust link between the AAAL and the AAAH could be used to retrieve the keying material. For instance in GSM when the MN is roaming, the BTS get a triplet (challenge, response, session key) from the Home Locator Repository of the "home network". The secret key of the MN is never transmitted on the network.

A2.4.9 Security References

[A2.28] [2-9] IETF draft "Mobile IP Authentication, Authorization, and Accounting Requirements", draft-ietf-mobileip-aaa-reqs-03.txt"

[A2.29] [2-10] P. Calhoun, A. Rubens, H. Akhtar and E. Guttman, "DIAMETER Base Protocol", draft-calhoun-diameter-16.txt, Internet Draft (work in progress), July 2000. (or http://www.diameter.org /)

[A2.30] [2-11] RADIUS C. Rigney, S. Willens, A. Rubens and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", RFC2865, June 2000.

[A2.31] [2-12] Durham, D., Boyle, J., Cohen, R., Herzog, S., Rajan, R., Sastry, A., "The COPS (Common Open Policy Service) Protocol", IETF RFC 2748, January 2000.

A2.5 Diversity Combining and Soft Handover Support

Certain air interfaces (e.g. UTRAN FDD mode) require or at least support the concepts of soft handover/macro diversity combining.

Essentially, this refers to the fact that a single MN is able to send and receive over two independent radio channels ('diversity branches') at the same time; the information received over different branches is compared and that from the better branch passed to the upper layers. This can be used both to improve overall performance, and to provide a seamless type of handover at layer 2, since a new branch can be added before the old is deleted.

Where diversity combining takes place entirely within the lower layers, this is not considered further here. (Sometimes, this is referred to as 'micro diversity'.) However, where diversity branches run through different BARs, a combining point must be available within the AN itself.

A2.5.1 Background

A hard handover is required where a mobile node is not able to receive or send traffic to two base-stations simultaneously. In order to move the traffic channel from the old to the new base-station the mobile node abruptly changes the frequency/timeslot/code on which it is transmitting and listening to new values associated with a new base-station. A good example is GSM where the mobile listens for new base-stations, reports back to the network the signal strength and identity of the new base-station(s) heard. When the old base station decides that a hand-over is required it instructs the new base-station to set up resources and, when confirmed, instructs the mobile to switch to a new frequency and time slot.

This sort of hand-over is called hard, mobile assisted, network initiated and backward (meaning that the old base-station is responsible for handling the change-over). Figure A2-22 shows a typical GSM hand-over.

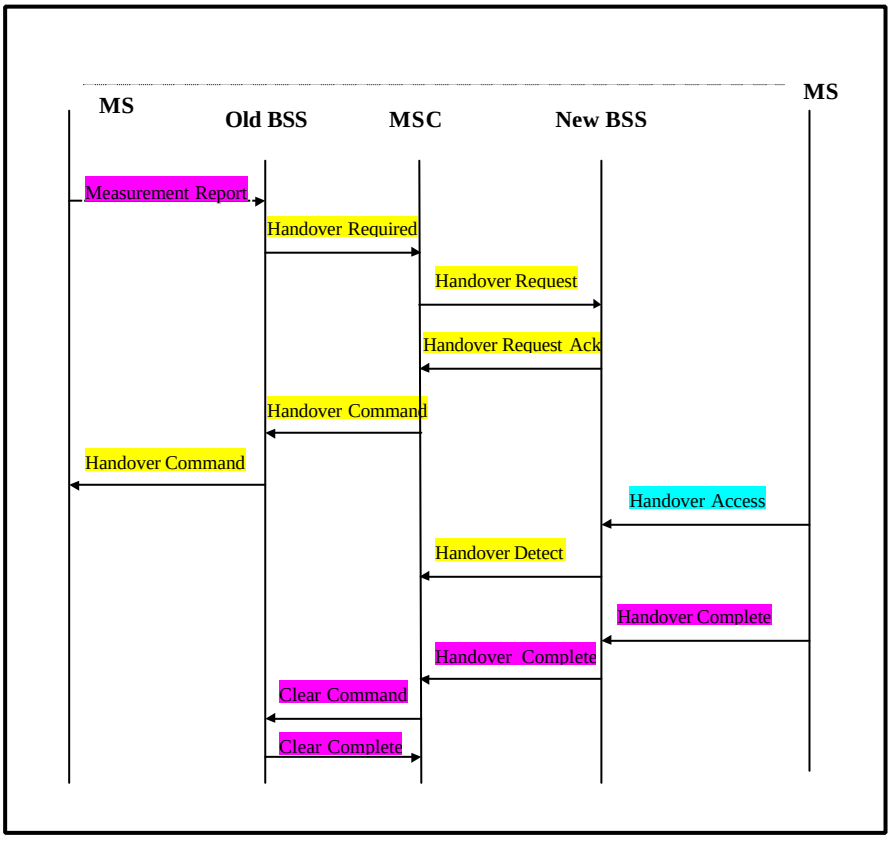


Figure A2-22: GSM Handover

A soft handover is possible when a mobile node can receive/transmit traffic/data to more than one base-station. A good example of soft handover is the UTRAN FDD mode. W-CDMA is particularly suited to soft handover because of the design of the receivers and transmitters: typically a rake receiver will be used to overcome the multi-path fading of the wide-band channel (Figure A2-23). Rake receivers have a number of so-called fingers, each effectively separate detectors, that are tuned to the same signal (i.e. spreading code) but delayed by different times. When the delay times are correctly adjusted and the

various components properly combined (this is micro-diversity combining) the effect of multi-path fading is removed. The rake receiver can also be used to detect signals from different transmitters by tuning the fingers to different spreading codes.

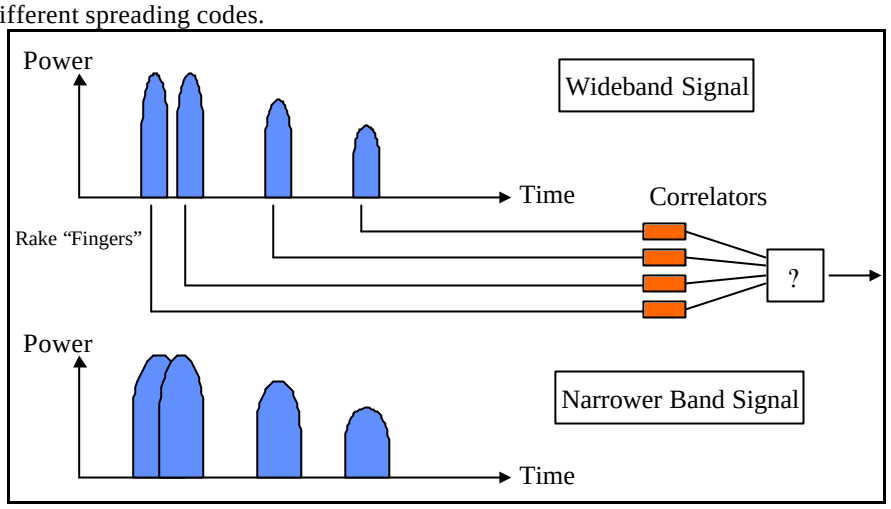


Figure A2-23: Diversity Combining

Soft hand-over is used in UTRAN FDD mode to increase capacity. In a TDMA system, such as GSM, the hard hand-over is delayed until the mobile has moved well within the coverage of the new base-station. If the handover threshold was set to the point where the new base-station signal exceeded the old then there would be a very large number of handovers as the mobile moved through the region between the cells and radio signals fluctuated – this would create a large signalling traffic. To avoid this, a large hysteresis is set – i.e. the new base-station must be (say) 10dB stronger for hand-over to occur. If the same was done in W-CDMA then the mobile would be transmitting a powerful signal to the old base-station and creating interference for other users – since in CDMA everyone else's transmissions are seen as noise– thus reducing capacity. To avoid this soft handover is used, giving an estimated doubling in capacity.

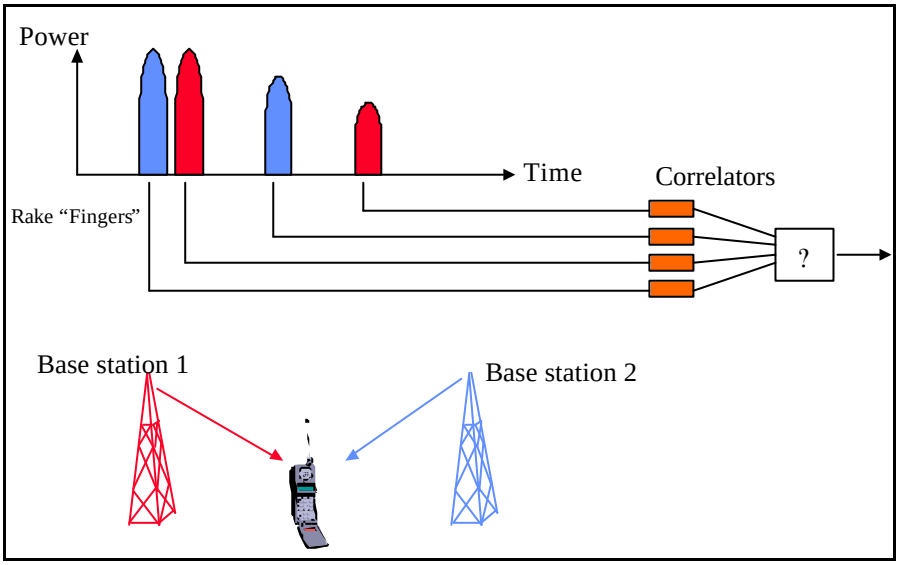


Figure A2-24: Soft Handover

A2.5.2 Macro and Micro diversity.

It is necessary to differentiate between combining/diversity that occurs layer 1/2 (physical and radio link layers), and that which occurs at layer 3 – the network layer. Here we use the term micro-diversity combining for the case where, for example, two antennas on the same transmitter send the same signal to a receiver over a slightly different path to overcome fading. As far as the network layer is concerned this is invisible.

Macro diversity takes place when the duplicating / combining actions take place over multiple base stations. This requires support from the network layer to move the radio frames between the base stations and a central combining point – the soft hand-over of UMTS described above falls into this category. In

the case of UMTS it is radio frames that are duplicated at some point in the network (the serving RNC) and sent to a number of Node Bs and, possibly via other (drift) RNCs. The combining that takes place at the serving RNC in the uplink direction is typically based on some simple quality comparison of the various received frames, which implies that the various copies of these frames must contain identical upper layer information. The serving RNC also has to do buffering to take account of the differing time of flight from each Node B to the RNC.

The term IP diversity is used to mean the splitting and combining of packets at the IP level. Now it can be argued that UTRAN diversity combining is really taking place at a lower layer, since it is radio frames that are being split and that these carry only fragments of the layer 3 PDUs (as well as other control information from within layers 1 and 2). Indeed, in the UMTS standards, diversity combining is modelled architecturally as a physical layer function even though it is placed at the serving RNC.

A2.5.3 Macro-diversity in BRAIN

The first point to make regarding soft handover and BRAIN is that we are unlikely to want to support soft handover for HIPERLAN/ 2:

?? HIPERLAN/2 uses OFDM and TDMA to divide the spectrum and so there will be little advantage, in terms of extra capacity, to providing soft handover. There may be some penalty to using a GSM – style hard-handover, but this appears the simplest solution.

?? HIPERLAN/2 transmitters and receivers will require significant modification to support soft-handover. A multiple branch receiver would be required for one. This would be a high frequency component and, as such, would likely push up the cost of the card. In addition we are aiming to only change the DLC and convergence layer of HIPERLAN/2 if at all possible.

Further study is needed by WP3 to determine if micro-diversity combining might help to overcome shadowing/fading in the kind of environments we envisage. The trade-off between the lost capacity taken up by the multiple signals and the gain by greater reliability needs to be investigated.

The next major issue is whether we would want to support other access technologies that require soft handover and macro-diversity. There are two possible approaches to this.

?? One approach is that this is a layer 2 issue and that the responsibility to split/combine and deliver with accurate timing the up and down signals is a layer 2 specific problem. After all, the network and lower layers are supposed to be independent but supporting soft handover at the network layer (IP) might require different timing tolerances/splitting points for different technologies. Imagine trying to support two different layer 2 technologies, both requiring soft hand-over support– they might require different timing tolerances and have different splitting/combining points, necessitating a relatively complex signalling protocol to set it up.

   This view that would say that adding UMTS to the BRAIN access network would be a case of taking IP down to the RNCs - i.e. the BRAIN Access Routers are RNCs – and leaving the macro-diversity to the specialised, ATM-based, RAN. There is no attempt to integrate the IP layer.

?? The second approach is to take IP all the way down to the transmitters – i.e. the BARs are the Node B equivalents and transmit directly with no layer 2 links between them. This is much more attractive in that it reduces the need for a specialised RAN and makes all the network elements just routers – the BARs just having radio cards. This would fit much better with the BRAIN access network design but is difficult technically as outlined in the following section.

A2.5.4 Why IP diversity combining is difficult

In this section we look at the difficulties and disadvantages of supporting soft handover at the IP layer in the BAN, this is further discussed in a recent IETF draft [A2.32]. In order to support soft handover at the IP layer several features would be required to be added to the BAN (Figure A2-25):

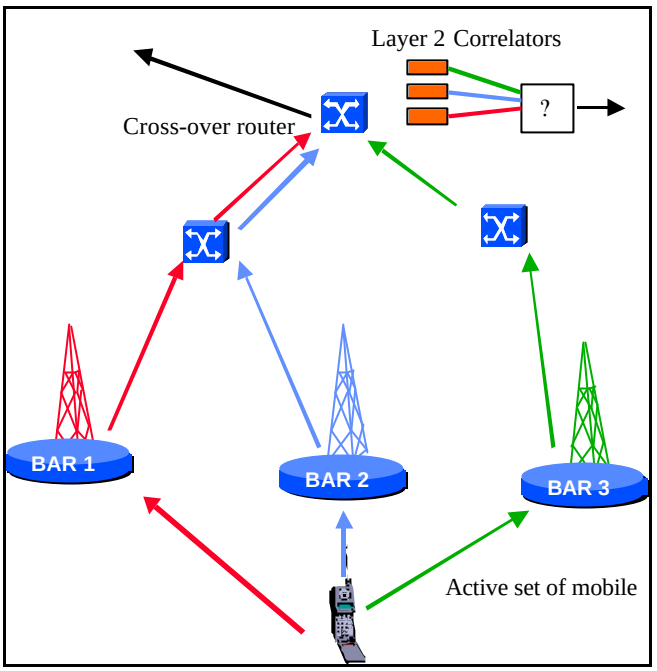


Figure A2-25: Possible Diversity Combining in the BRAIN

Synchronised timing: One of the requirements of the diversity combining is that the different bit streams arrive simultaneously at the combining point, in the network or terminal, typically within 5-80ms in UTRAN, and the relative jitter between the various streams is very low. In the UTRAN FDD mode the radio frames are carried by ATM to avoid introducing jitter. Using IP an ATM-like jitter guarantee would be required and something like IntServ would be needed to set up the priority paths. Buffers would be needed at the mobile, BWRs and the cross-over router to allow this synchronisation. In addition there would have to be a synchronised time across the BAN and a mechanism for discovering the delays across the various hops in order to deliver the synchronised bit streams at the combining points. It might well be possible to integrate this information within a micro-mobility management protocol. For example if this was tunnel based – like Mobile IP – the timing and quality information could be included within the encapsulation header or in an extension header, in the way that IPSec Authentication Headers are added.

A splitting protocol: In the downlink direction there is a requirement for a router to duplicate packets and send them to multiple destinations – the BARs in the active set of the mobile. No current routing protocol, with the exception of multicast protocols, achieve this. Running a local multicast routing within the BAN is possible, indeed it is one of the proposed micro-mobility protocols, but, for example HAWAII and Cellular IP would require modifications. Even running a local multicast there are issues such as how fast the protocol can converge. Note that it would not be necessary to have a multicast group per terminal – a multicast address could be associated with a particular group of BARs and the host associated with this particular set. This approach would probably be easier to integrate with some micro-mobility protocols.

Traffic differentiation: In order for diversity combining to work all traffic (that requires soft handover) must be treated as real time – even best effort IP packets must be delivered from the splitting point to the terminal with minimal jitter. This is obviously will decrease network efficiency but if it is not done then the best effort traffic will contribute a very significant noise increase – and capacity decrease – in a CDMA system like UMTS. However, note that the network inefficiencies in Node B – RNC communication (i.e. the requirement for all traffic to be low jitter and real-time) are inherent in the diversity combining requirement and nothing to do with the use of ATM or IP per se.

A combining protocol: The multiple up-streams of IP packets, from the various base-stations in the active set, must rendezvous at a router somewhere in the BAN, be recognised as part of the same transmission and all sent to a layer 2 entity for combining. The difficulty is in deciding where the cross over router should be and in recognising, on a per packet basis, that all these streams are really part of the same transmission. If they are sent from the same IP address and port to the same destination IP address and port then that may be good enough. The rest of the BAN must also ensure the "duplicate" packets arrive simultaneously, a sequence number will probably be needed, and that the jitter is very low.

There is a further difficulty in the case where the link layer operates with backward error correction – in that case the uplink data stream influences what gets sent on the downlink and you then have to synchronise up and down link layers as well.

A signalling mechanism: Finally a protocol will be needed to link the BWRs involved in active set of the mobile and the cross-over router. This protocol would be responsible for measuring the delays between the BWRs and the crossover point and adjusting the transmission time of packet to ensure simultaneous arrival at either the crossover router or mobile. This could be combined with a micro-mobility protocol, e.g. local multicast, to offer generic soft and hard hand-over support

Introducing these elements into the BAN will result in several major disadvantages:

?? The micro-mobility routing protocol becomes very complex.

?? A large amount of state is stored in the network.

?? A complicated synchronisation mechanism is required.

?? A low jitter QoS mechanism must be provided – even if this is not otherwise required.

This looks a major overhead and it is currently recommended that the basic BAN architecture does not attempt to support soft handover. Options for incorporating some kind of soft handover support might be considered at a later stage.

A2.5.5 Diversity References

[A2.32] J. Kempf, P. McCann, P. Roberts, "IP Mobility and the CDMA Radio Access Network: Applicability statement for Soft Handoff", IETF draft, <draft-kempf-cdma-appl-00.txt>, July 2000.

A2.6 Radio Resource Management

In classical (GSM/UMTS) terminology, the problem of radio resource management includes primarily the following subjects:

?? Deciding whether to allocate a (radio) channel to a terminal, taking into account current cell loading.

?? Carrying out the negotiation to allocate that channel to the terminal.

?? Deciding (on the basis of relative neighbour cell measurements, or in order to balance resources between different cells or within the resources of a single cell) to modify the radio channel allocated to a terminal – including as a special case handover.

Note that handover between system types (air interface technologies) can be considered under this heading, although the detailed procedures in this case are usually slightly different.

These radio resource management requirements, although originally elucidated in the context of 2nd and 3rd generation networks, are in fact generic to any mobile wireless network. They therefore have to be supported in the overall BRAIN network also. However, in the overall BRAIN architecture, it is a design goal to maintain a clean separation between network related issues (which should be generic to any air interface) and issues specific to a given air interface. In BRAIN system, the network layer has to take part in any procedure that involves handovers between BARs (also called network layer handovers), since this requires re-routing and possibly network-related QoS reconfiguration within the wired access network. Therefore, these actions cannot take place entirely below the IP_2W interface, which also implies that the decision making activities must take place above this interface.[16] This makes radio resource management at least partially the concern of the network layer.

This section presents a model architecture for decoupling air-interface specific aspects of radio resource management from the problem of handover (network assisted terminal mobility), which is a necessary part of designing a BAN which is generic to a set of different air interfaces. This architecture then forms part of the boundary between the generic network layer, and the air interface specific support:

?? The network layer provides information to the radio resource management (RRM) function, and responds to commands from it.

?? The RRM function operates air interface specific algorithms, which may also be tailored to a specific operational environment.

Note that the RRM function may require an interface to the traffic engineering/management functions within the wired access network, and indeed to QoS brokers within the IP core, in order to complete the admission control decision. These aspects are considered as part of the end-to-end QoS problem; this section relates to the radio specific parts only.

A2.6.1 Protocol Architecture

Radio resource management requires the network (and in some architectures the MN) to maintain a picture of the current utilisation of the radio resource, and to have a well defined control point where allocation decisions can be made.

According to the above decomposition of the interface between the network layer and RRM function, we can consider two groups of message exchanges.

?? **Information Reporting**

Reporting of radio measurements from the BAR to the RRM
Reporting of radio measurements from the MN to the RRM
Reporting of resource requests (made by the MN) from the BAR to the RRM

?? **Control**

Allocation of radio resources by the RRM to a pair (MN, BAR)
Initiation of a handover of an MN between two BARs

Note that there is no assumption about the physical location of the RRM function, and indeed there are several options. There could be a centralised RRM within a BAN (gathering information from the BARs); there could be an RRM which operates in a distributed fashion among the BARs themselves; and most likely there will be an RRM within each mobile node as well. Different RRMs may fulfil different parts

[16] Strictly, this only applies to the BAR, but we take the same picture in the MN for simplicity. In any case, procedures taking place entirely below the IP_2W, such as intra-BAR handover are not considered here. Note that this includes radio channel reconfiguration carried out in order to re-arrange the resource allocation within a cell.

of the function; for example, an RRM in the network may control resource allocation, while RRMs in the MN may initiate handovers. The following architecture does not constrain any of these choices, and all can be supported simply by re-routing messages and responses. (The only restriction is that we do not support reporting of measurements *to* mobile nodes.[17]) Where the BAR reports information remotely, and accepts remote decisions about resource allocation, the messages would naturally be carried over its IP terrestrial links.

With this in mind, we can outline the following message flows at the MN and BAR as in Figure A2-26. Note that flows over the air interface are shown only as primitives at the IP_2W level (and in fact, only at the IP_2W control interface); these may be implemented over the air either as IP packet flows, or specialised layer 2 flows. The choice does not affect the rest of the discussion, and may indeed be different for different air interfaces.

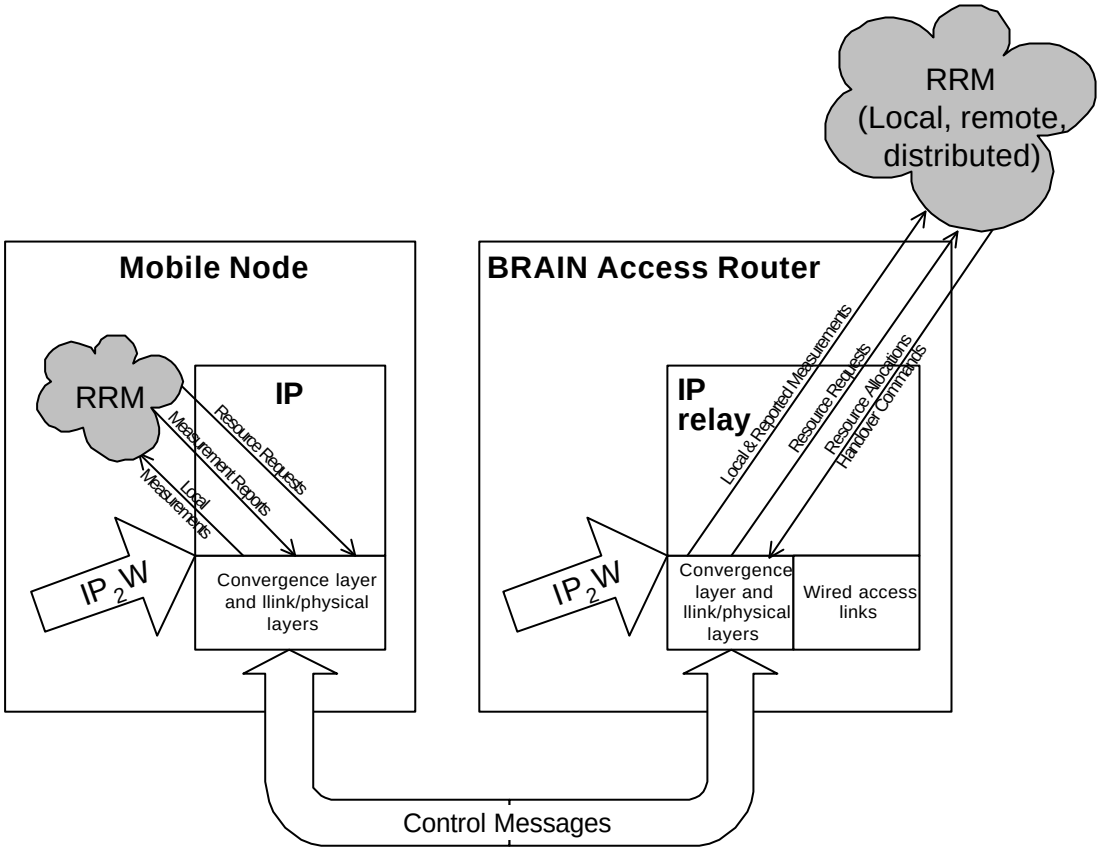


Figure A2-26: Radio Resource Management Message Exchanges

The architecture diagram shows local measurements and measurement reports being passed between the MN and BAR, and maybe from the BAR to a remote RRM server. In the current basic form of the IP_2W interface, these are assumed to be expressed in some general, air interface neutral form, which makes a this generic architecture very easy to implement. However, it should be noted that there is no fundamental reason why the content of the messages could not be made specific to the air interface, since the service interface and message passing function just relay these measurements transparently between the actual measurement function within the physical layers and the RRM function itself. This might be required for some air interfaces in order to achieve maximum efficiency.

[17] This reflects the current situation in 2nd and 3rd generation systems, and might have to change in an ad hoc network. However, it is a reasonable initial restriction in the current concept.

A2.7 Multicast

This section lays out an architecture for multicast support in future BRAIN networks. In the context used in this document, multicast support refers to the ability of hosts inside BRAIN networks to participate in existing multicast sessions and initiate new ones. A multicast session is enabled by the standard IP multicast mechanisms and identified at the network layer by an IP multicast group address. The initial perspective used in this document is a MN's one, meaning the requirements will first be applied to indicate what MNs expect and are required to perform in multicast situations. The assumption is that once a preliminary model is established, other perspectives such as a router's one (either a BAR or a wired network router) would easily be deduced. This and similar documents are intended to clarify and solve issues related to enabling IP multicast for MNs inside BRAIN networks and are not related to support for terminal mobility via deployments of multicast mechanisms.

A2.7.1 Introduction to Multicast

The IP multicast group model is as follows: multicast groups are identified in IPv4 by Class D group addresses; and in IPv6 by the address prefix FF00::/8. Group membership is anonymous and receiver initiated, i.e., a host may join a group by just "tuning in" to the group address[18]. After joining a group members may send to or receive packets from other group members. In some cases senders need not be group members. The multicast service extends the IP service through the join and leave messages for initiating and terminating group membership. These messages each carry a multicast group address as parameter. Semantically the data transfer is unreliable best effort and connectionless. In IPv4, the Internet Group Management Protocol (IGMP) [A2.33] is used for information exchange between hosts and multicast routers on a physical subnet. When a host joins a new group it sends IGMP messages to the respective multicast routers from where multicast routing can be established. Routers make a routing table entry and periodically poll their associated subnet or hosts. The corresponding protocol for IPv6 is Multicast Listener Discovery (MLD) [A2.38], which is derived from IGMPv2. MLD uses ICMPv6 (IP Protocol 58) message types, rather than IGMP (IP Protocol 2) message types.

In the link layer, mapping of IP multicast to broadcast based link layers is usually a straightforward solution: Ethernet and FDDI support unicast, multicast and broadcast addresses, while Token Rings have functional addresses to reach group of receivers.

Multicasting can be split into three major tasks:

?? **Addressing**

This is self-explanatory and involves management of IP Class D addresses or IPv6 multicast addresses.

?? **Multicast Initiation**

There are two distinct parts of this task. The first part is discovering present groups-to-sessions mappings either for joining or sending packets to the group. The actual mechanisms for discovering group-to-sessions mappings are outside the scope of this document but an example method is the look-up of distributed session directories. The second part of Multicast Initiation is related to hosts that intend to join discovered multicast groups and need to contact the nearest multicast routers, which are responsible for the routing of multicast traffic. This part is achieved by IGMP (or MLD). IGMP is deployed between multicast routers and hosts, which are assumed to be on the same subnet (A2.7.2) but can alternatively achieve virtual link-layer connectivity through tunnels. There are three released versions of IGMP specifying the mechanisms for exchange of control messages. The newer versions of IGMP are intended to reduce the leave latency (defined as the time needed before a multicast router realises that a host is no longer interested in the multicast group) by inclusion of new messages.

?? **Multicast Routing**

This task is concerned with setting up of routes and forwarding packets to group members sharing a common multicast address. This is performed by the multicast routing protocols such as Distance Vector Multicast Routing Protocol (DVMRP) [A2.34], Multicast Open Shortest Path First (MOSPF) [A2.35], Protocol Independent Multicast – Dense Mode (PIM-DM) [A2.36], PIM – Sparse Mode (PIM-SM) [A2.36] and Core Based Trees (CBT) [A2.37]. These protocols have been designed for IPv4, and some accommodation is needed when they are applied for IPv6. For example, [A2.39] outlines recommendations in the use of PIM to support IPv6. The most

[18] This is not a totally passive operation, since the host may need to send group join messages to force the multicast data to be transmitted on the local link.

commonly used classification criteria is mainly concerned with the scalability of multicast routing protocols and divides multicast routing protocols into two categories: dense and sparse mode routing protocols. The main difference between sparse and dense mode multicast protocols is that dense mode protocols use variations of broadcasting (flooding) to distribute packets to group members and sparse mode protocols use a centre-point router called Rendezvous Point (PIM-SM) or Core (CBT), to which sources send packets and interested hosts explicitly join.

A2.7.2 Mobility and Multicast

Multicast routing protocols are designed to cope with dynamic group membership but not with dynamic group member location. Thus a problem arises when MNs are using some of the mentioned multicast routing protocols, since there is a need to re-establish routing trees after handovers. The main challenge is to reduce delays during the re-computation of multicast delivery trees and thus reduce packet losses when a mobile group member crosses cell boundaries during a multicast session.

This problem is relevant both for the mobile sources because the new base station (multicast capable) has to find a route to the multicast delivery tree, and also for the mobile receivers where there is a join and graft latency caused by the new base station's subscription to the multicast tree. Mobile IP is taking into account multicast support and proposes two solutions: remote subscription and bi-directional tunnelling. Remote subscription is a straightforward method where MNs in foreign networks simply subscribe to the multicast group and form delivery trees to their current location. This model provides a simple solution but imposes problems in IPv4 due to the incorrect source address of multicast packets sent by MNs.

In Mobile IPv6, the use of the care-of address as the IP source address in conjunction with the Home Address option allows the home address to be used but still be compatible with multicast routing that is based in part on the source address.

For the bi-directional tunnelling MNs are receiving and sending packets through Home Agents via unicast IP tunnels. This solution hides mobility of hosts from the multicast group but creates some routing overhead and for some cases sub-optimal routing. This would render the use of multicast invisible to the BAN.

There is a significant contribution to the problem of providing multicast for MNs in the Internet research community. The schemes expand on the initial solution presented by Mobile IP and propose more complex ones. It turns out that there are several different detailed approaches for where in the access network multicast capabilities are supported, and which one is followed has a significant effect on the detailed requirements for multicast support on the BAN external interfaces. These issues are therefore covered in more detail in chapter 3.

A2.7.3 Multicast References

[A2.33] W. Fenner, "Internet Group Management Protocol, version 2 (IGMPv2)", Internet Working Draft, '98.

[A2.34] D. Waitzman, S. Deering and C. Partridge, "Distance Vector Multicast Routing Protocol", RFC 1075, '88.

[A2.35] J. Moy, "Multicast routing extension for OSFP", Commun. ACM, vol. 37, no. 8, pp. 61-66, Aug. '94.

[A2.36] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, C. Liu and L. Wei, "Protocol Independent Multicast (PIM): Protocol Specification", Internet Draft, Jan 11. '95.

[A2.37] A. Ballardie, "Core Base Trees (CBT version 2) Multicast Routing ", RFC 2189, '97.

[A2.38] S. Deering, W. Fenner, and B. Haberman, "Multicast Listener Discovery (MLD) for IPv6", RFC 2710, October 1999.

[A2.39] B. Haberman, H. Sandick, and G. Kump, "Protocol Independent Multicast Routing in the Internet Protocol Version 6 (IPv6)", Internet Draft (work in progress), draft-ietf-pim-ipv6-03.txt, March 2000.

A2.8 Location Based Service Support

This section discusses aspects of a positioning service for location-aware applications, states requirements, examines existing approaches and presents an initial architecture for such a service. The terms used in this section follow the IETF Spatial Location Working Group notion.

To support location-aware applications, a secure and scalable service is needed that allows such applications to ask for the current positions of a user. The components of such a service can be divided as follows. The *positioning service* is responsible for answering location requests of clients about the position of targets. The *location mechanism* is responsible to determine the position of targets and to deliver that information to the positioning service. The components of the positioning service can be described as follows (see Figure A2-27).

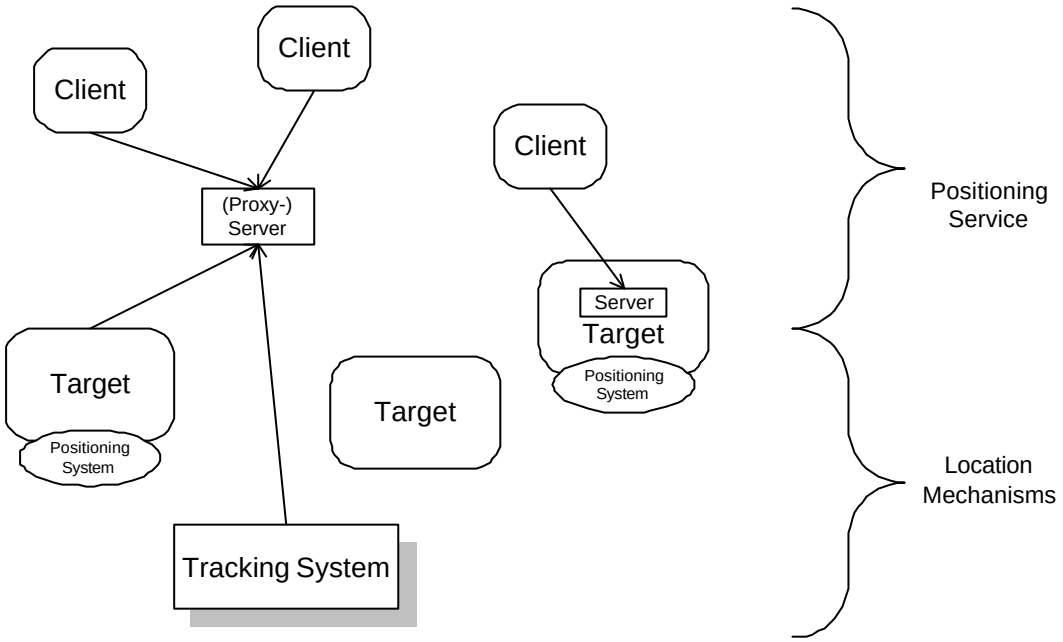


Figure A2-27: Components of the Positioning Service

?? **The Client**: This is the element that requests the Physical Location of something (called the Target).

?? **The Server**: This is the element that provides the Physical Location of the Target to the requesting Client. This Server could either be a process on the Target Device, or a Proxy Server representing the Target.

?? **The Target**: This is the element whose Physical Location the Client requests.

?? **The Proxy Server**: Is a Server which either aggregates great numbers of Targets for location requests, or represents itself in place of Targets which do not have their own Server Processes.

The elements of the location mechanism consists can be divided into two types**. Positioning systems** are attached to a mobile object; they are able to determine their own position. **Tracking systems** monitor distinct areas and are able to determine the position of mobile objects in this area. These two types can also be subdivided into *cellular systems* that consists of (overlapping) areas, and *non-cellular systems* which determine the exact position within a certain system with a certain accuracy.

A2.8.1 Application scenarios

The following small application scenarios illustrate different query types that can be answered by a positioning service.

?? Free time: Shopping and Hobbies

In this scenario, Sandy enters a shopping mall. At a point, she wants to go to a specific shop. The navigation component of the shopping mall application determines the current position of Sandy and presents her a route to the desired target.

In this scenario, the request is of the type “location-of-an-object-query”. The tracked target itself is interested on the location information.

?? Free time: Shopping and Hobbies II

Sandy still wants to go to a specific shop. Unfortunately, she takes the wrong lift. As the navigation component tracked Sandy on her way to the shop, the component reroutes her the correct way.

As in the first scenario, the request is of the type "location-of-an-object-query", and the tracked target itself is interested on the location information. But as the navigation component knew about Sandy's intention to find the shop, the component decided to let the positioning service send events in order to track her route.

?? Medical care

In this scenario, Mr. Pangalos' health is monitored by a small device connected to a BRAIN network. In case of a medical problem, the hospital is informed about the problem. Then the hospital supervision application would ask the positioning service for the location of Mr. Pangalos in order to send help if the problem is severe.

In this scenario, the request is also of the type "location-of-an-object-query". But now it is not the tracked target itself that is interested on the location information, but an authorised third party.

?? Nomadic worker

In this scenario, Stephanie Jones intends to attend a meeting for which she has to take a plane in order to get at the meeting place. As the flight had a delay, she's a little bit late, so in the taxi from the airport she asks the meeting application whether all participants (except her) are already there. The application asks the positioning service for the objects in the meeting room and compares the result with the expected list of names.

In this scenario, the request is of the type "objects-at-a-location-query".

A2.8.2 Requirements

We will now describe the requirements needed for an appropriate positioning service. These requirements are stated in more formal terms in the requirements section of the deliverable.

1) A query asks for the position of a specified user.

   This is the most basic query and needed for a lot of application scenarios, e.g. the ones used in WP.1.

2) The service should support different locating mechanisms.

   As in reality, a whole number of location mechanisms may exists, that work even in parallel (on one location, GPS can be used and the location mechanism of the wireless network the user is currently using, on another location, only a cell-based infrared beacon system exists), the system should not have a problem with this heterogeneity.

3) The position service should be not tied to any individual network type.

   Therefore, it has to be located at the application layer, since no network-specific mechanisms must be used.

4) The service should conceptually work even when the target is not connected to the access network.

   In case a tracking system is used to determine the position of the target, there is no need for a network connection of the target. Therefore, the service should be able to work also in this case (if only a positioning system can be used, a connection is of course essential).

5) A central user management cannot be assumed.

   In a global environment like the Internet, there is no central authority that manages or authenticates users (a key certificate is not that same as managing a user). Therefore, the system cannot rely on such an assumption, especially not on a "transfer-of-trust" from the system.

6) The service should not have a "Single-Point-of-Failure"

   i.e. the service should not depend on the existence of a single node.

7) The service has to be scalable

   in order to be used on a global scale.

8) The service has to be secure

   Especially location information is very sensitive data. If the user does not feel comfortable with a service that knows his/her location, nobody will use it. Additionally, there are legal aspects that require a sensitive treatment of this aspect. The general outline has to be that the tracked user has to be able to control the access to the location data.

A2.8.3 Non-Requirements

To start with a simple, manageable positioning service that already allows a lot of application scenarios, it has been decided to omit the following three requirements.

1) Objects-at-a-location-queries
   As this requirement would require mechanisms to find all servers, which requires a much more complex system design.

2) Events
   Events add convenience for the programmer to the system, additionally, they can reduce network load under certain circumstances. As events do not require a change in system design of the positioning service, we decided to omit them for now, but to re-evaluate this decision later on.

3) Fault-tolerance
   It seems acceptable that a small amount of objects cannot be located or the location information is expired if a node fails or a network partition occurs.

A2.8.4 Existing approaches

A2.8.4.1 IETF Spatial Location BOF

There exists an IETF "Birds-of-a-feather" group [A2.40] that has already started work. Its aim is to develop a protocol (called "Spatial Location Protocol (SLoP)") that allows "[...] an application on an IP network acquire the location of something represented on an IP network, in a reliable, secure, and scalable manner" [A2.43]. It is intended to convert the BOF group to a regular IETF Working Group (WG). By June, 2001, a first version of the protocol is intended to be finished.

From the current state of the discussion, SLoP would rather well implement the above requirements, even if some design decisions have not been determined yet (e.g. the question of whether Pull, Push, or both should be supported).

SLoP is based on IP, but it is not decided yet whether TCP or UDP will be used. SLoP does explicitly not define the way a position is determined by the positioning system.

A2.8.4.2 Nexus

Currently being implemented as a part of Nexus [A2.42], a research project that targets an open global platform for location-aware applications, the "Distributed Universal Location Service" [A2.41] aims at realising a positioning service as well. As the focus in this project lies on scalability and fault-tolerance, this work emphasises the replication aspects. So position data is stored not only on primary, but also on a number of secondary servers, so clients are able to access more local position information (which might be coarser due to privacy reasons). This positioning service additionally takes into account "objects-at-location-queries" so the design is a little bit more complex than the IETF approach. Finally, this service supports events, so a client does not need to pull every request from a server, but can be actively informed by a server (push).

A2.8.4.3 Discussion

As the SLoP proposal seems to fit into the above requirements, and since it can be expected that the IETF proposal will become a standard in the Internet world, it is therefore suggested to realise a positioning service that uses the SLoP protocol. The way in which this fits into the overall BRAIN network layer architecture is shown in Figure A2-28.

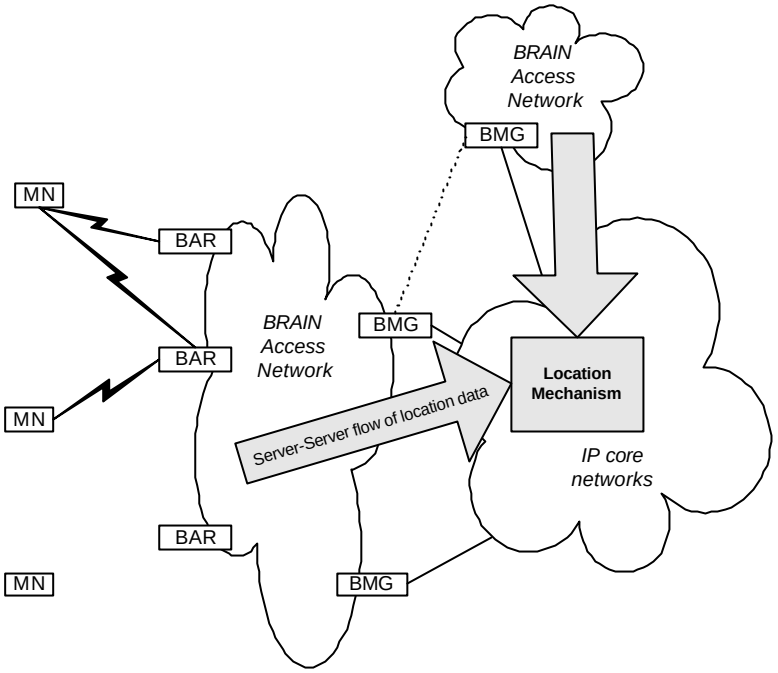


Figure A2-28: SLoP In the BRAIN Architecture

As SLoP does not define a location mechanism, the mobility and radio resource management functions of BRAIN can be used as a base on which a location mechanism (in this case a tracking system) can be realised. This location mechanism could be located outside of BRAIN Access Network, for example in a BRAIN core network or at any other authorised node in the Internet. It would probably use information from the access network. The access network manages the micro mobility of the terminal which can be used to track the terminal or user position, see picture above. Due to the cell-based nature of BRAIN, the position information offered by the BRAIN location mechanism could consist of messages for example of the form *(ObjectId,Position,Area,"Cellular",TimeStamp)* where *ObjectId* is the id of the tracked object (e.g. assigned IP address), *Position* determines a (WGS84-)position, e.g. the centre of the cell, *Area* determines the area of the cell, and *TimeStamp* a time stamp in order to determine the validity of the message (see [A2.41] for details).

In order to complete the SLoP world, BRAIN could propose a SLoP extension that allows positioning mechanisms to communicate with servers.

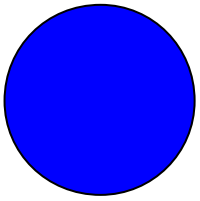
A2.8.5 Location Based Services References

[A2.40] http://www-nrc.nokia.com/ip-location/

[A2.41] A. Leonhardi, U. Kubach: An Architecture for a Universal, Distributed Location Service, Proceedings of the European Wireless '99 Conference, Munich, Germany, pp. 351-355, ITG Fachbericht, VDE Verlag, 1999.

[A2.42] http://nexus.informatik.uni-stuttgart.de/

[A2.43] "Problem and Requirements Statement" for the Spatial Location Protocol (SLoP) BOF within the IETF 'Applications Area'. http://www-nrc.nokia.com/ip-location/PR-Statement-20000502.txt



A3 Mobility Management Annex

A3.1 IETF Handover Protocols and BRAIN Handover Design

A3.1.1 Introduction

This annex examines the existing IETF proposals for support of seamless handover. The handover survey has been used for extracting requirements and solution ideas to the design of the handover portion of the overall BRAIN micro-mobility protocol and to the BRAIN wireless convergence layer ("IP to Wireless Convergence Interface", IP_2W). The surveyed handover schemes have been suggested for use in the context of host mobility management based on Mobile IP and its variants for regional mobility. However, we assume that, although the original context is in proxy agent architectures, the handover schemes can be adapted to other classes of mobility protocols as well.

We look into the proposals for fast handover frameworks and specific solutions. The Generalized IP Handoff proposal [A3.2] has been used as a reference architecture because it is one of the few attempts to investigate the handover procedure at a conceptual level without attaching itself to a specific mobility protocol and without going into detailed syntax of the protocol messages. The generalised handover messaging is further elaborated in "EMA Enhanced Mobile IPv4/IPv6"[A3.3], which specifies the actual handover signalling protocol using "Mobile IPv6"-like messages. In the mean time, the Mobile IP Working Group has assigned design teams for finding a consensus in selecting the most promising solutions for fast handovers in IPv4 and IPv6, whereas the newly created Seamoby WG investigates context transfer between access routers (among other things). The focus is on the proposed IPv6 solutions.

Based on the handover protocol investigations, we define the basic functional requirements for the BRAIN handover protocol and describe the detailed design issues, which need to be solved. Finally, the proposed protocol is illustrated and briefly compared with the existing solutions.

A3.1.2 Generalised Handover Framework

The Generalized IP Handoff proposal [A3.2] specifies the signalling between the MN and the old and new access routers (OAR and NAR[19], respectively). It is generic in two aspects: it tries to cover many possible handover scenarios and it leaves the message formats open. The main characteristics of the framework are:

?? the handover is mobile controlled – optionally network assisted and constrained
?? the handover may be planned or unplanned (proactive or reactive)
?? the connectivity to access routers may be of type make-before-break or break-before-make
?? the framework enables the MN to be isolated from the mobility routing (horizontal signalling vs. vertical signalling).

The framework proposal separates the protocol into two parts:

a) handover preparation (forward protocol) and
b) handover completion (reverse protocol).

These could be viewed, respectively, as

a) the first part of a planned handover and
b) the latter part of a planned handover which coincides with an unplanned handover.

However, an unplanned handover consists of negotiations between OAR and NAR that are performed in the handover preparation phase of the planned handover. Therefore, a planned handover that fails can be restarted at the NAR as an unplanned handover. Nevertheless, for clarity we examine complete planned and unplanned handovers separately (and we do not regard the reverse protocol as a continuation of the forward protocol).

[19] Along with OAR and NAR, this document uses the term CAR to denote a candidate access router that subsequently may serve the MN as its NAR (aka SAR).

A3.1.2.1 Planned Handover

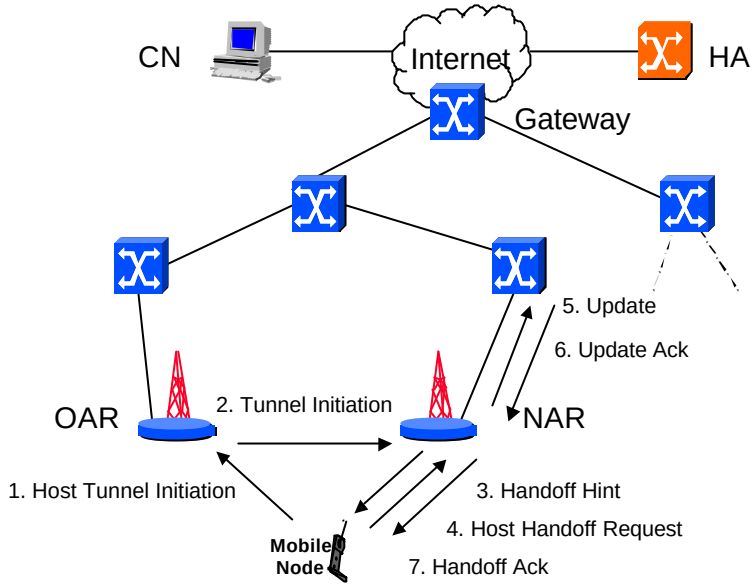


Figure A3-1: A Generalised Planned Handover

The following signals can be identified in a planned handover (see Figure A3-1[20]):

1. Host Tunnel Initiation (H-TIN)
   Request from the MN to OAR to prepare a handover to NAR. If the handover is possible, OAR builds a forwarding tunnel to NAR. Otherwise it return a Handoff Denial message (not shown in the figure). The Initiation message may contain a request for replicated data transfer (bi-casting).

2. Tunnel Initiation (TIN)
   Conveys MN's state information to NAR. On arrival of this message NAR establishes the tunnel for "IP diversity". This message may be sent to several CARs. The message may convey the MN's context information (e.g., credentials) to the NAR.

3. Handoff Hint (HH)
   An optional indication from NAR to MN of the NAR's readiness to handover. The MN may receive these hints from several CARs, and the message may contain performance and preference information for the MN to assist in ranking the CARs.

4. Host Handoff Request (H-HR)
   Request from MN to NAR to initiate handover. The NAR authenticates the request.

7. Handoff Ack (HAck)
   Handover Acknowledgement from the NAR to MN.

5b. Update (UPD)
   A vertical path update message of the local mobility protocol.

6. Update Ack (UPDAck)
   A vertical path update acknowledgement message of the local mobility protocol.

[20] Note that in the figures, path updates are directed to an upstream router. However, depending on the network topology or the mobility protocol scheme, the path updates may also be addressed to the OAR (for example, if OAR is the anchor ns or the routing protocol is MER-TORA). Note also that OAR and NAR might not be physically adjacent (i.e., they do not necessarily share a common link in the wired network).

A3.1.2.2 Unplanned Handover

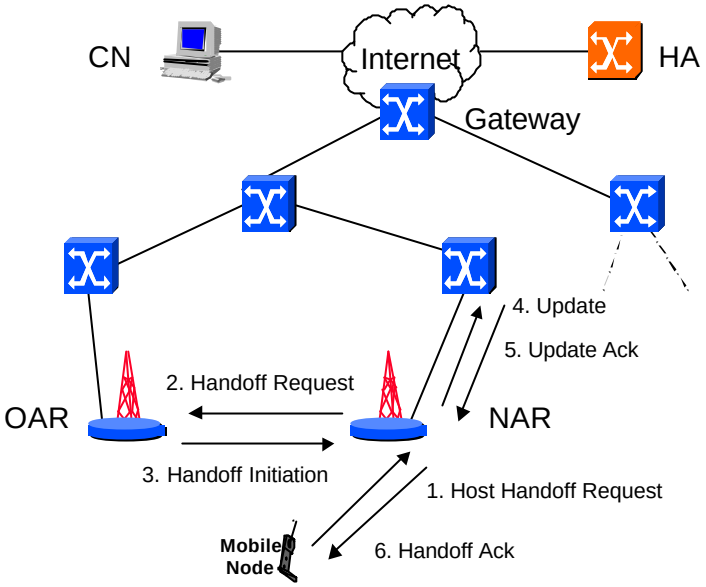


Figure A3-2: A Generalised Unplanned Handover

The reverse protocol of the unplanned handover (see Figure A3-2) introduces the following additional messages:

2. Handoff Request (HR)
   Conveys MN's credentials to be checked at the OAR (unless NAR can perform authentication locally) and requests for a forwarding tunnel from the OAR.

3. Handoff Initiation (HI)
   Conveys MN's state information to NAR. On arrival of this message NAR establishes the tunnel for "IP diversity". This message may convey the MN's context information to the NAR. If the MN is not authorised, Handoff Denial message is sent instead.

The generalised handover framework has been further developed in [A3.3], which uses enhanced Mobile IPv6 signalling between the MN and access routers.

A3.1.3 Proposed Handover Schemes

A3.1.3.1 Introduction

This section introduces several handover schemes that have been proposed in the IETF. The schemes presented here are:

?? EMA Enhanced Mobile IPv6/IPv4 [A3.3] <draft-oneill-ema-mip-00.txt>
?? A Framework for Smooth Handovers with Mobile IPv6 [A3.7] <draft-koodili-mobileip-smoothv6-00.txt>
?? Fast Handovers in Mobile IPv6 [A3.9] <draft-koodli-mobileip-fastv6-01.txt>
?? Foreign Agent Assisted Hand-off [A3.14] <draft-calhoun-mobileip-proactive-fa-02.txt>
?? Fast Handoffs in MIPv6 [A3.6] <draft-elmalki-handoffsv6-01.txt>
?? Fast Handovers for Mobile IPv6 <draft-designteam-fast-mipv6-00.txt>

Except for [A3.14] , these schemes are designed for IPv6. The [A3.6] scheme has an equivalent IPv4 counterpart ([A3.5]), which along with [A3.14] forms the pair of competing proposals for IPv4. The IPv4 Handoff design team of the Mobile IP Working Group in the IETF has not been able to merge the proposals or reach consensus on the superiority of the solutions.

Other fast handover proposals that are not considered further in this document include

?? Real-time Mobile IPv6 Framework, which mainly addresses AAA and QoS aspects in a planned handover but it also suggests proactive operation using Neighbour Discovery Redirect similarly to [A3.9].

A3.1.3.2 EMA Enhanced Mobile IPv6/IPv4 (EMA-MIP)

The EMA Enhanced Mobile IPv6/IPv4 proposal [A3.3] introduces MIPv6-type signalling for the Generalized IP Handoff framework (that is, Destination Options are used for carrying the messages during handover). Although TORA is assumed as the EMA:MER routing protocol, the handover protocol is largely isolated from TORA specific path updating. The draft is extensive in also describing inter-domain handover scenarios. However, here we only look into planned and unplanned intra-domain handovers.

Although the whole idea of EMA is to assign a semi-static co-located CoAs (EMA-CoA) to a MN for its visit in a domain, EMA-MIP expects the MN to acquire a temporary CoA (nCCoA) at each NAR for "horizontal" signalling. The need of this temporary address partially ruins the idea of avoiding address allocations and it may be a disadvantage in terms of performance and protocol complexity.

The path updating protocol is not fully transparent in MER-TORA because the "tau" value (a factor in the node's "height") is transferred during horizontal signalling.

Unplanned Handover

The signals for an unplanned handover are shown in Figure A3-3.

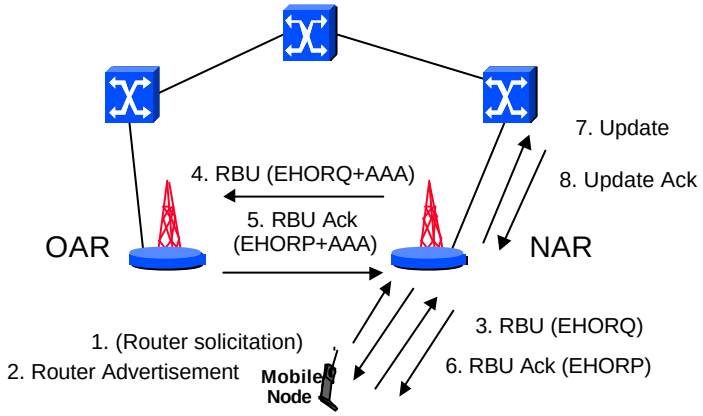


Figure A3-3: EMA-MIP Unplanned Handover

The messages are defined as follows:

3. RBU = Reverse Binding Update (Host Handoff Request)
   Sent by MN to OAR via NAR and containing EMA Hand-over Request Destination Option (EHRQ) and a Routing Header. NAR authenticates the request.

4. RBU = Reverse Binding Update (Handoff Request)
   RBU sent by MN with AAA Request Option added by NAR. Requests installation of forwarding from the semi-static EMA-CCoA to MN's temporary nCCoA.

5. RBUAck (Handoff Initiation)
   Confirms the binding (EMA-CCoA, nCCoA) and provides NAR with status of the handover in EMA Handover Response (EHORP) Destination Option and policy information in AAA Response Destination option. Forwarded to MN via NAR using Routing Header.

6. RBUAck (Handoff Ack)
   RBUAck sent by OAR via NAR to acknowledge the RBU.

Depending on the path updating scheme and OAR's capability to authenticate MN's rerouting request, NAR may send the Update and RBU signals in parallel or NAR may need to wait for an acknowledgement from OAR before sending the Update.

Planned Handover

The messages sent in a planned handover a shown in Figure A3-4.

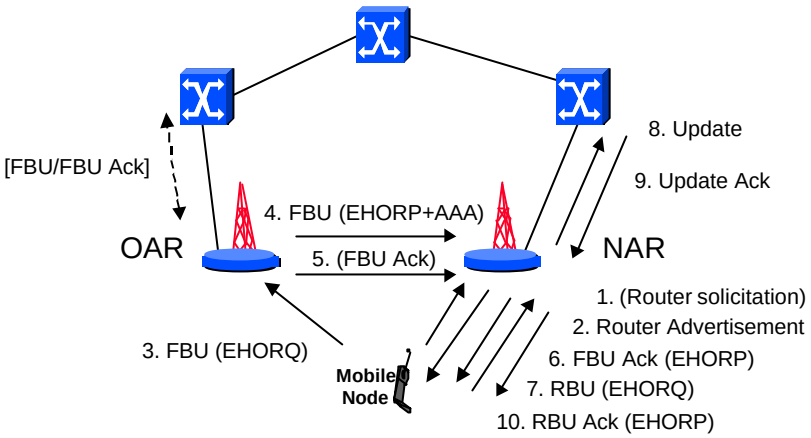


Figure A3-4: EMA-MIP Planned Handover

In the planned handover, the NAR's address (and the MN's future nCCoA) must be known while the MN is connected to the OAR. Therefore, the MN must have either L3 or L2 connectivity to NAR if it assists in the handover.

The MN must be able to form nCCoA at NAR for use as the source address in horizontal signalling.

The messages are defined as follows:

3. FBU = Forward Binding Update (Host Tunnel Initiation)
   Sent by MN directly to OAR and containing EMA Hand-over Request Destination Option (EHRQ) and a Routing Header. Requests installation of forwarding from the semi-static EMA-CCoA to MN's temporary nCCoA. The MN may suggest handover to several NARs. OAR authenticates this message.

4. FBU = Forward Binding Update (Tunnel Initiation)
   FBU sent by MN with AAA Response Option added by OAR.

5. FBUAck
   Confirms the binding (EMA-CCoA, nCCoA).

6. FBUAck (Handoff Hint)
   FBUAck sent by NAR to acknowledge the FBU.

If the MN does not receive FBUAck, it can proceed as in an unplanned handover (i.e., by sending an RBU message). If the handover is initiated by a network controller, the MN is not aware of the FBU/FBUAck signalling. Therefore, there should be a means for signalling the MN of the handover.

A3.1.3.3 A Framework for Smooth Handovers with Mobile IPv6

This framework [7] specifies envelopes for transferring the MN's state between OAR and NAR. The framework supports mobile controlled (unplanned) and network controlled (planned) handovers. The handover request and unsolicited handover reply messages are authenticated with IPSec and authentication data. OAR maintains the MN's state for a while before purging it. Context transfer request implicitly includes a request for a forwarding tunnel from OAR to NAR for a small amount of time.

This framework is used for buffer management in [A3.10] and for transferring header compression state information in [A3.12].

The signalling in the context transfer framework is shown in Figure A3-5.

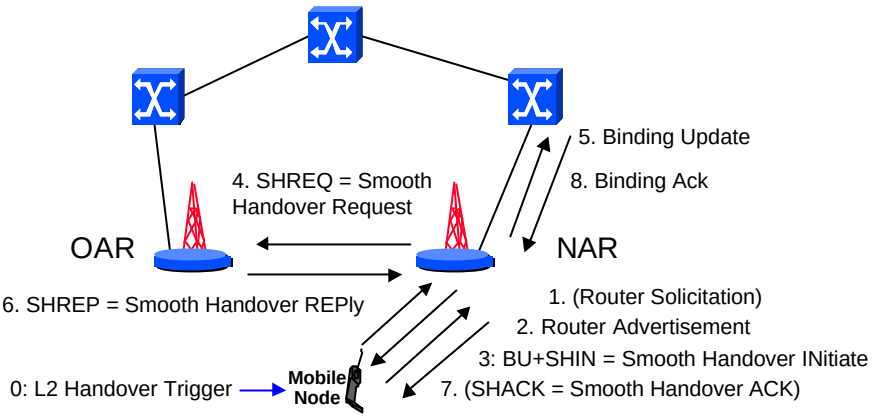


Figure A3-5: Framework for Smooth Handovers (unplanned MCHO)

The context transfer framework uses the following messages in addition to the standard Router Solicitations and Advertisements:

3. SHIN (Smooth Handover INitiate)
   Used by the MN to request for selected "smooth handover features" (such as buffering or header compression state transfer). This message has a tailored authentication data (i.e., not using IPSec). This is also an implicit request for setting Binding Cache entry at OAR for the MN's new CoA.

4. SHREQ (Smooth Handover Request)
   The requested features in SHIN are forwarded from NAR to OAR.

6. SHREP (Smooth Handover REPly)
   Transfers MN's context information from OAR to NAR.

7. SHACK (Smooth Handover Ack)
   Optionally acknowledges MN's context transfer requests.

In proactive operation (NCHO), OAR can send an unsolicited SHREP (without a SHREQ from NAR). Proactive operation is shown Figure A3-6. If OAR does not know MN's new CoA, NAR may need to provide this information by sending a SHREQ.

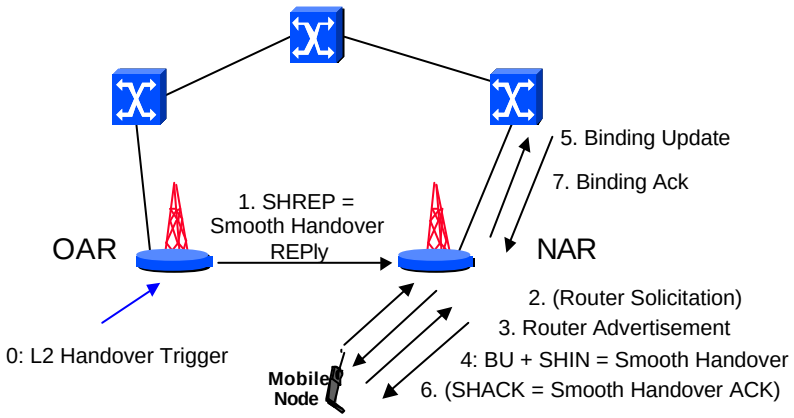


Figure A3-6: Framework for Smooth Handovers (planned NCHO)

Recently, this framework has been updated to "Context Transfer Framework for Seamless Mobility" [A3.8] by removing dependencies from Mobile IPv6 and taking into account the Mobile IPv6 Handoff Design Team's work (see Section A3.1.3.7). In the new proposal, which is targeted to the IETF Seamoby WG, also planned mobile controlled operation is specified by adding a "P-SHIN" (proactive SHIN) message from the MN to OAR. This revised framework is used for buffer management in [A3.11] and for transferring header compression state information in [A3.13].

A3.1.3.4 Fast Handovers in Mobile IPv6

The Fast Handovers in Mobile IPv6 proposal [9] aims at reducing latencies in NAR identification, address acquisition, and packet forwarding in a network controlled handover. It uses the Neighbour Discovery Redirect message to instruct the MN to move to NAR. The Redirect message contains the link-local and link-layer address of NAR, and information for the MN to form a new CoA at NAR. Then, OAR supplies the MN's Interface Identifier and/or new CoA to NAR and instructs it to act as ND proxy for the address. This message (either ICMP or unsolicited SHREP) also transfers other context (e.g., security keys) and sets up a forwarding tunnel towards NAR. If the proposed new CoA is not unique in OAR, it returns a Handover Error message to NAR. The MN authenticates itself (e.g., using SHIN message) and sends a Binding Update in the same packet using encapsulation.

The signalling in this fast handover scheme is shown in Figure A3-7.

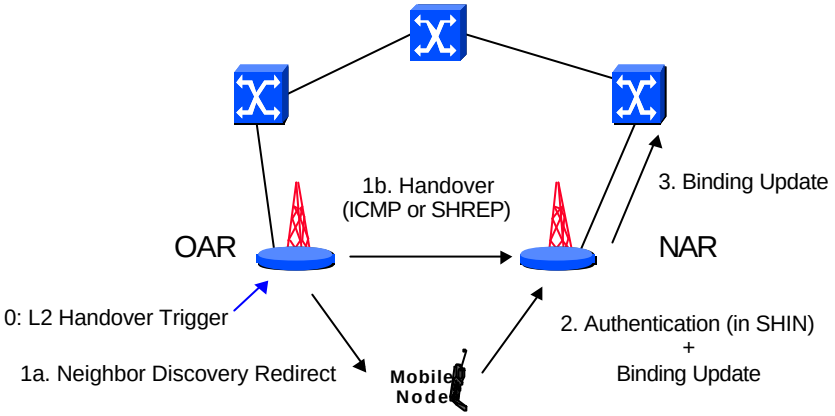


Figure A3-7: Fast Handovers in Mobile IPv6

A3.1.3.5 Foreign Agent Assisted Hand-off

In Foreign Agent Assisted Hand-off [A3.14], which is an IPv4 scheme, NAR registers on behalf of the MN. To avoid the round trip of agent solicitations/advertisements, this scheme relies on link-layer indications of imminent handovers. These indications, called triggers, may occur at OAR (source trigger) or at NAR (target trigger). This scheme is intended for use with Regional Registrations, but it is independent of the routing topology; that is, it supports regional registrations for both gateway-FAs and anchor-chained FAs.

To avoid needless registrations when the MN moves back and forth between ARs, this scheme suggests adding hysteresis to agent advertisements and using bi-casting of downstream traffic from the gateway or anchor FA via both OAR and NAR to the MN.

Source-Triggered Handover

The source-triggered handover can be roughly classified as a network controlled planned handover. Figure A3-8 illustrates a handover that is triggered at OAR.

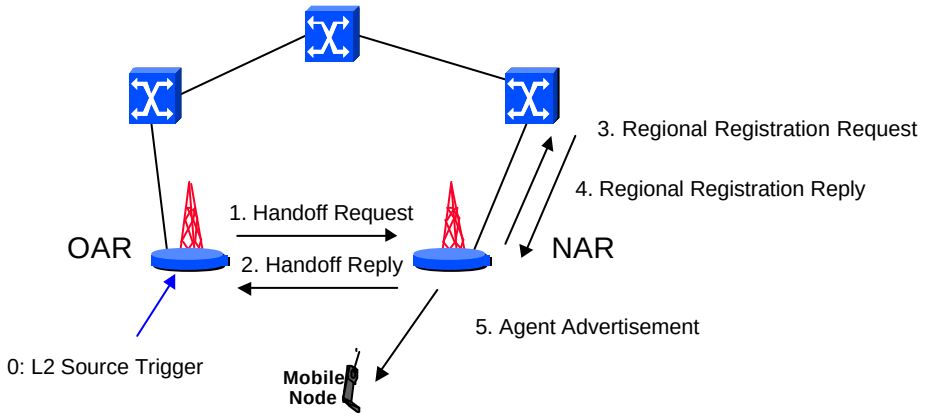


Figure A3-8: FA Assisted Source-Triggered Handover

The Handoff Request from OAR to NAR carries MN's Home Address and its link layer address, Home Agent's address, GFA's IP address (if any), remaining registration lifetime, and security information. Using this information NAR proactively registers on behalf of the MN.

Target-Triggered Handover

The target-triggered handover can be roughly classified as a network controlled unplanned handover. Figure A3-9 illustrates a handover that is triggered at NAR.

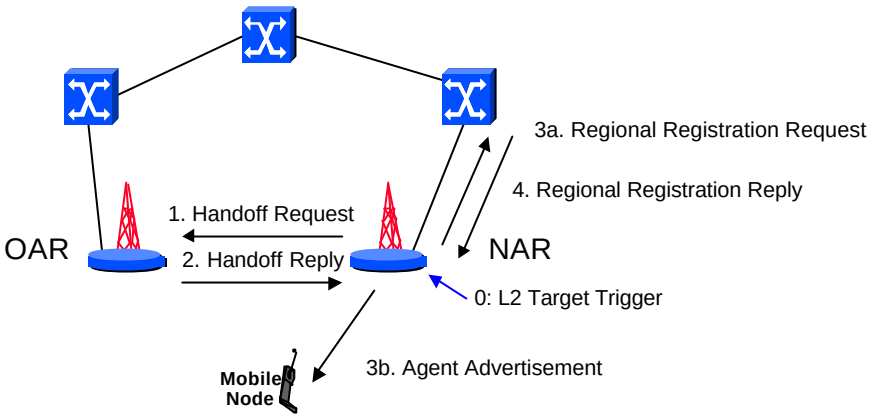


Figure A3-9: FA Assisted Target-Triggered Handover

If GFA is used, NAR needs to know the GFA's IP address in order to make the surrogate registration. The Handoff Reply from OAR to NAR carries MN's Home Address and its link layer address, Home Agent's address, GFA's address (if any), and security information. A Handoff Request is only needed unless the link-layer provides the information needed for the registration.

This scheme was one of the MIPv4 Fast Handoff Design Team's proposals, which has been merged with its contender [5] (see Section A3.1.3.6) in [A3.19].

A3.1.3.6 Fast Handoffs in MIPv6

The Fast Handoffs in MIPv6 proposal [A3.6], which is a straightforward adaptation of the corresponding IPv4 scheme[A3.5], builds on two key ideas that are orthogonal: a) registering with NAR through OAR and b) using bi-casting to avoid packet loss.

The signalling in this fast handover scheme is shown in Figure A3-10.

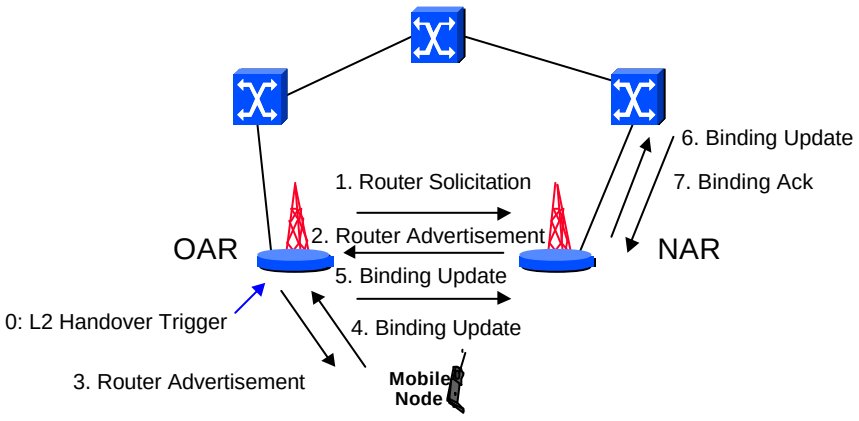


Figure A3-10: Fast Handoffs in MIPv6

This scheme assumes that the MN is able to perform a registration via OAR before the L2 connection to OAR is closed. The OAR solicits NAR for a Router Advertisement, which may need routing between different subnets. Alternatively, the advertisement could be conveyed using L2 messaging between access points at OAR and NAR. This would need tight coupling to L2 procedures, which is clearly a disadvantage. The proposal also speculates on MN requesting NAR advertisement via OAR by sending a Tunnel Initiation message in the Generalized IP Handoff framework.

The MN may request for its MAP (Mobility Anchor Point) to bi-cast downstream traffic to both OAR and NAR. In a flat architecture the OAR may be the anchor point (i.e., MAP).

This proposal relies heavily on the availability of the link to OAR in preparing for the handover to NAR, which makes it vulnerable to an abrupt loss of connection. On the other hand, it is economical in its reuse of already existing protocol messages.

A3.1.3.7 Fast Handovers for Mobile IPv6

The Fast Handovers for Mobile IPv6 proposal [A3.16] is an output of the Mobile IPv6 Handoff Design Team. It only considers planned handovers without make-before-break capability. There are variations on the basic scheme depending on whether the handover is network or mobile controlled, what link-layer capabilities are assumed, and how a new CoA is generated for the MN.

The signalling for a network controlled handover is shown in Figure A3-11.

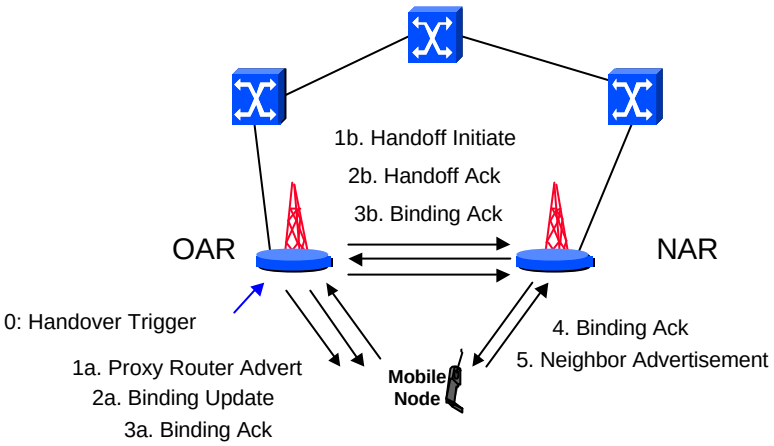


Figure A3-11: Fast Handovers for Mobile IPv6 (Network Controlled)

In the network controlled handover, OAR advertises NAR in a proxy Router Advertisement and assists the MN in generating a new CoA at NAR. The MN selects NAR based on advertisements that includes NAR's prefix (or the new CoA), and sends to OAR a Binding Update, which includes its new CoA. The MN may request an acknowledgement for the Binding Update. The acknowledgement is bi-casted through both OAR and NAR and it is mandatory if NAR needs to perform Duplicate Address Detection on the suggested new CoA. OAR should bi-cast the acknowledgement to NAR.

If stateful address autoconfiguration is used, the Handoff Initiate/Ack handshake is performed before the Proxy Router Advertisement can be sent.

The scheme supports buffering and bi-casting from OAR to NAR during the handover.

The signalling for a mobile controlled handover is shown in Figure A3-12.

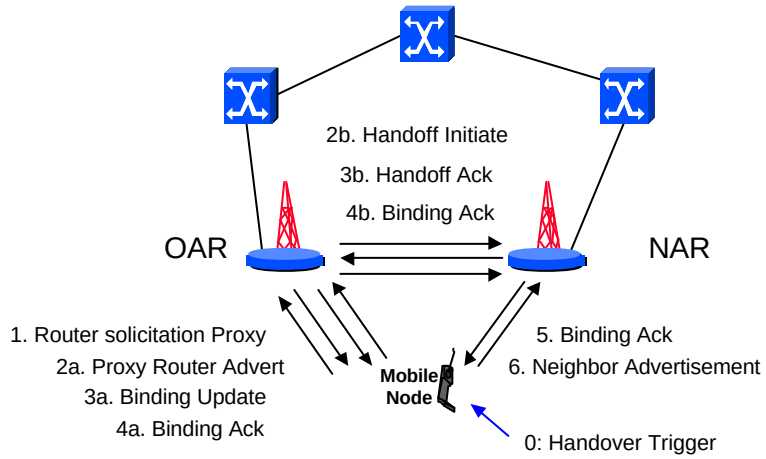


Figure A3-12: Fast Handovers for Mobile IPv6 (Mobile Controlled)

In the mobile controlled handover, the MN first sends a Router Solicitation Proxy message that indicates a desired new point of attachment. This identifies a link-layer element that may be connected to OAR. Otherwise the protocol is the same as in the network controlled handover.

A3.1.4 Comparison of the Handover Schemes

A3.1.4.1 Basic Characteristics

The individual fast handover schemes can be analysed in terms of whether they

?? support planned and/or unplanned handovers
?? support mobile controlled and/or network controlled handovers
?? rely on specific link-layer support (e.g., MBB or link-layer "triggering")
?? forward or bi-cast from OAR or an anchor point in order to prevent packet loss
?? allow for context transfers between OAR and NAR
?? optimise signalling over the wireless link
?? are basically independent of the regional mobility protocol

The basic properties of the schemes are summarised in Table A3-1.

	Planned/ Unplanned	**MCHO/ NCHO**	**expected L2 support**	**forwarding/ bicasting**	**context transfer**	**signals over wireless link**	**coupling with MIP**
1) EMA Enhanced Mobile IPv6/IPv4	P/U	MCHO / NCHO	concurrent connectivity to OAR and NAR in planned MN-assisted HO	tunnelling from OAR to NAR	tau, addresses, AAA	(Router Solicitation), (Router Advertisement), FBU, FBUAck, RBU, RBUAck	none
2) A Framework for Smooth Handovers with Mobile IPv6	P(NCHO)/ U(MCHO)	MCHO / NCHO	HO trigger at NAR or MN	tunnelling from OAR to NAR	generic container	(Router Solicitation), (Router Advertisement), SHIN, SHACK	MN sends a BU using encapsulation
3) Fast Handovers in Mobile IPv6	P	NCHO	HO trigger at NAR	tunnelling from OAR to NAR	generic container	ND Discovery Redirect, BU	MN sends a BU using encapsulation
4) Foreign Agent Assisted Hand-off	P/U	NCHO	Source Trigger at OAR or target trigger at NAR	bi-casting from OAR or anchor	MN+GFA+HA addresses, registration lifetime AAA	Agent Advertisement	NAR register using Regional Registration
5) Fast Handoffs in MIPv6	P	NCHO	HO trigger at OAR	bi-casting from anchor	-	Router Advertisement, BU	MN registers using a form of Regional Registration
6) Fast Handovers for Mobile IPv6	P	MCHO / NCHO	optional triggers at OAR or MN	bi-casting or tunnelling from OAR	NAR's prefix and L2 address, MNs old/new CoA, Home Address and L2 address	(RtSolPr), PrRtAdv, BU, 2*BUAck, BU/NA	BU to OAR and BU to Home Agent may be combined

Table A3-1: Basic Properties of the Handover Schemes

Note that Router Solicitations and Router Advertisements are not necessarily integral parts of the handover schemes but they are the means for the MN to discover NARs (i.e., they are part of movement detection). Some schemes do not take these into account but they loosely refer to the corresponding link layer support (i.e., L2 triggers).

All schemes allow for setting up a forwarding tunnel for downstream packets from OAR to NAR. This functionality is also part of the MIPv6 base protocol and it is suggested as part of route optimisation for MIPv4 [[A3.1].

A3.1.4.2 Protocol Messages

The following tables summarise the messages used in the planned and unplanned handover schemes (partially adopted from [A3.2]). The number of the handover scheme refers to the scheme numbering in Table A3-1. The messages are mapped against the message templates specified in Generalized IP Handoff framework. The messages in parentheses do not semantically correspond to the framework messages but they could be extended to include the needed semantics.

Scheme	TIN	H-TIN	HH	H-HR	H Ack	UPD	UPD Ack
1	FBUAck	FBU (EHORQ)	FBUAck	RBU	RBUAck	UDU	UDUAck
2	SHREP		(Router Adv)	SHIN+BU	SHACK	BU	BUAck
3	SHREP or ICMP			SHIN+BU		BU	BUAck
4	HOReq		(Agent Adv)			RegReq	RegRep
5	(BU)					BU	BUAck
6	HI	RtSolPr		(NA)	BAck	BU	BUAck

Table A3-2: Messages in the Handover Schemes mapped to the Handover Framework (Planned)

Scheme	H-HR	H Ack	HR	HI	UPD	UPD Ack
1	RBU	RBUAck	RBU	RBUAck	UDU	UDUAck
2	SHIN+BU	SHACK	SHREQ	SHREP	BU	BUAck
3						
4		(Agent Adv)	HO Req	HO Rep	Reg Req	Reg Rep
5						
6						

Table A3-3: Messages in the Handover Schemes mapped to the Handover Framework (Unplanned)

An acknowledgement message for “Tunnel Initiation” is missing from the Generalized IP Handoff framework, whereas such message is defined for Foreign Agent Assisted Hand-off and for Fast Handovers for Mobile IPv6.

Only EMA-MIP and Fast Handovers for Mobile IPv6 provide a “Host Tunnel Initiation” message from the MN for kicking off a planned handover. Fast Handoffs in MIPv6 suggest a planned scheme that significantly deviates from the framework. Fast Handovers in Mobile IPv6 that only considers planned handovers uses ND Redirect message for informing the MN of the existence of NAR.

A3.1.4.3 Solutions to Design Issues

All of the examined protocols share the following design patterns that can be used as guidelines for any handover design:

?? packet loss is avoided by forwarding and/or bi-casting MN’s downstream data from OAR to NAR
?? the urgent downstream path diversion happens at OAR – path updating at routers that reside farther away from the MN is not time-critical
?? time-critical acquisition procedures (e.g., DAD and AAA) are avoided by pushing or pulling MN’s state from OAR to NAR (alternatively the MN could convey this information)
?? CAR selection is out of scope of the handover protocol

For unplanned handovers none of the fast handover schemes adds much value to the smooth handover feature of basic Mobility Support for IPv6, where the previous router may act as a temporary home agent for the MN. The only addition in the proposed new schemes is the possible provision for transferring context features between access routers.

With respect to planned handovers the following observations about handover preparation, handover decision, message authentication, address acquisition, and coupling with path updates, can be made by looking into the existing proposals:

1. Handover preparation and location of handover control

   All schemes except for EMA-MIP lack the network-layer signal from the MN to OAR for suggesting preparations for handover to CARs (i.e., Host Tunnel Initiation). The existence of this signal alone

can be viewed as an indication that the handover is "mobile controlled" (or more precisely "mobile initiated"). However, all schemes that allow the MN to perform the final registration decision can be claimed to be mobile controlled. Therefore, of these schemes only "Foreign Agent Assisted Hand-off" is fully network controlled. It does not give the MN any means for controlling the handover at the network-layer. Nevertheless, it does not force the MN to register with the network-selected NAR either.

Furthermore, "link-layer triggers" are hidden from the network layer, which means that any scheme that relies on "source-triggering" (i.e., triggering at OAR) could alternatively be defined as mobile-initiated (and possibly network-constrained) by assuming triggering at the MN and the necessary link-layer messaging by which the MN informs OAR of CARs. However, a generic (network-layer) solution should not depend on such link-layer messaging.

2. Determination of NAR at OAR and the MN

The Generalized IP Handoff framework emphasises that the MN controls the handover but the framework does not specify how the MN acquires information about CARs. That is, the framework lacks the network-layer signal that advertises CARs to the MN. The following approaches can be recognised for acquiring the address of NAR:

Scheme	NAR Determination at OAR or MN
1) EMA Enhanced Mobile IPv6/IPv4	MN is simultaneously connected to OAR and NAR, and MN listens to NAR's Agent Advertisements. MN sends this information to OAR in "H-TIN".
2) A Framework for Smooth Handovers with Mobile IPv6	(Refers to the mechanisms of "Fast Handovers in Mobile IPv6")
3) Fast Handovers in Mobile IPv6	OAR magically gets the identification of NAR, which OAR advertises to the MN using ND Redirect message.
4) Foreign Agent Assisted Hand-off	OAR magically gets the identification of NAR through link-layer messaging that is out of scope of the specification. OAR does not send this information to MN.
5) Fast Handoffs in MIPv6	MN solicits or gratuitously receives Agent Advertisements (that OAR has solicited) through OAR. NAR determination at OAR is out of the scope of the specification.
6) Fast Handovers for Mobile IPv6	OAR magically learns about candidate ARs, which OAR advertises to MN in Proxy Router Advertisements.

All schemes except for EMA-MIP, which assumes the MN's simultaneous connectivity to OAR and NAR, leave the responsibility of NAR determination to unspecified link-layer procedures. Therefore, without the availability of a make-before-break connection the MN would not know about available CARs through network-layer signalling.

Only in EMA-MIP, the MN can explicitly select CARs by sending a message to OAR. The network controlled schemes assume that MN will move to (one of the) CAR(s) and perform the proactive inter-AR signalling and path updating anyway.

The Generalized IP Handoff framework, and accordingly also EMA-MIP, defer the final decision on NAR at the point where CARs advertise their availability and their offered service level to the MN while other schemes fix the selection of (a single) NAR earlier or leave the issue open.

3. Identification of the MN at NAR

In order for NAR to reserve resources, authenticate, or to perform proactive path updating it must know the IP address or other identification of the MN. The following approaches for conveying the identification of MN can be recognised:

Scheme	MN Determination at NAR
1) EMA Enhanced Mobile IPv6/IPv4	MN registers with NAR while still having the connection to OAR.
2) A Framework for Smooth Handovers with Mobile IPv6	OAR pushes MN's old CoA to NAR in the SHREP message.
3) Fast Handovers in Mobile IPv6	OAR pushes MN's new CoA to NAR in the Handover message (either ICMP or SHREP).
4) Foreign Agent Assisted Hand-off	OAR pushes MN's Home Address, link-layer address, and Home Agent's address to OAR in the Handover Request message.
5) Fast Handoffs in MIPv6	This is NCHO. NAR does not need MN's CoA before MN connects to NAR.
6) Fast Handovers for Mobile IPv6	OAR sends MN's old and new CoA, and link-layer address to NAR in the Handoff Initiate message.

Only Foreign Agent Assisted Hand-off and Fast Handovers for Mobile IPv6 specify a two-way handshake between access routers for conveying MN-information.

4. Message authentication

The following key generation and authentication mechanisms have been specified:

Scheme	Key Generation and Message Authentication
1) EMA Enhanced Mobile IPv6/IPv4	MN is assigned a Mobile ID, which may be IPv6 address or NAI, for example. MN and the access network share a Private Authentication Key (PAK)=MD5(MID, network key), and the MN and NAR share Private Identification Key (PIK) =MD5(nCCoA, network key)
2) A Framework for Smooth Handovers with Mobile IPv6	Messages between access routers maybe authenticated using IPSEC AH. MN's messages are authenticated with SHREQ Authentication Suboption.
3) Fast Handovers in Mobile IPv6	OAR pushes MN's session keys to NAR. The Handover message contains an ICMP Handover Authentication Suboption (see 2) above).
4) Foreign Agent Assisted Hand-off	Generic Key Reply Extension can be used for distributing MN's session key. FA's have pre-established or dynamic security associations. FA-FA Authentication extension is used between FAs.
5) Fast Handoffs in MIPv6	Not specified (but could use IPv6 equivalent of MIPv4 Generalised Key Reply extension for delivering the session key among routers)
6) Fast Handovers for Mobile IPv6	Refers to the case of Mobile IPv6 for security requirements.

EMA-MIP uses a key generation algorithm that has been adopted from Cellular IP [A3.17].

5. Address acquisition

Depending on the micro-mobility protocol, the MN's CoA may or may not change when the MN changes an access router within an access network. Each time a new CoA is acquired its uniqueness in the scope of its use should be verified. Therefore, some means of performing Duplicate Address Detection (DAD) should be specified. In the various schemes, the following addressing acquisition procedures can be recognised:

Scheme	Address Management
1) EMA Enhanced Mobile IPv6/IPv4	The MN retains its Co-located CoA ("EMA CCoA") across intra-domain handovers. The MN acquires a new CoA ("nCCoA"), which is it uses for horizontal signalling with access routers. OAR can determine the new CoA by knowing NAR's prefixes and MN's MAC address. DAD is not addressed.
2) A Framework for Smooth Handovers with Mobile IPv6	MN performs DAD using Neighbor Discovery or OAR may somehow know the new CoA using a mechanism that is not specified.
3) Fast Handovers in Mobile IPv6	OAR informs the MN of NAR's prefixes in the ND Redirect message, which allows the MN to construct its new CoA. Alternatively, OAR may construct a new CoA for the MN. OAR transfers the new CoA to NAR that starts acting as a proxy for the address if it is unique.
4) Foreign Agent Assisted Hand-off	The MN uses FA-CoAs.
5) Fast Handoffs in MIPv6	The MN can construct the new CoA based on NAR's prefix in its Router Advertisements received via OAR. In a MAP domain, MAP takes care of uniqueness of CoAs by notifying the MN that the Interface Identifier is not unique. In a flat architecture, the MN may use the new CoA while simultaneously performing DAD.
6) Fast Handovers for Mobile IPv6	OAR may assign a new CoA to MN or MN may generate a new CoA based on NAR's prefix. If the new CoA is statelessly generated NAR may perform DAD on behalf of MN.

6. Path updating

Coupling with path updating have an impact on how easily a handover scheme can be adapted to various micro-mobility schemes. Most of the handover schemes are intended for use with MIP v4/v6

or their extensions for localised operation. Therefore, tight coupling with MIP can be justified. The schemes have been integrated with path updating as follows:

Scheme	Integration with path updates
1) EMA Enhanced Mobile IPv6/IPv4	NAR sends UDU to OAR and receives UDUAck from OAR. This sets a host route at OAR and triggers BUAck to MN.
2) A Framework for Smooth Handovers with Mobile IPv6	MN sends BU with SHIN. NAR forwards the BU to HA or to a regional agent after consulting with OAR (SHREQ/SHREP).
3) Fast Handovers in Mobile IPv6	MN sends BU with SHIN. NAR forwards the BU.
4) Foreign Agent Assisted Hand-off	NAR sends a Regional Registration on behalf of MN.
5) Fast Handoffs in MIPv6	MN sends a BU via OAR and through NAR.
6) Fast Handovers for Mobile IPv6	MN sends normal MIP binding updates. However, BU to OAR and BU to HA might be combined in the same IP packet.

Obviously, if several CARs are available, path updates should not be performed before the MN has selected its NAR and connected to it. In contrast, in network controlled handovers (e.g., in "Foreign Agent Assisted Hand-off") with a predetermined NAR, path updates can be performed before the MN has established a link to NAR.

A3.1.5 BRAIN Handover Protocol Design

A3.1.5.1 Scope and Functional Requirements

A handover specification should provide solutions to the high-level functional design issues identified in the previous section. In the following, the design issues are further elaborated in order to contrive the requirements for the events and information flows that constitute the handover protocol.

The overall mobility support involves the following type of operations:

?? initial access to the network (including AAA procedures, which may involve authorisation with servers outside the BAN, and address allocation)
?? inter-domain handover (that reduces to initial registration if there is no specific support for handovers across domains)
?? intra-domain handover (with accelerated authorisation and address allocation (if any) procedures)
?? de-registration (which is only needed if resources (such as addresses) have to be explicitly released)

This section only addresses intra-domain handovers between access routers[21] by finding answers to the functional issues. The identified basic functional requirements for the seamless handover protocol are as follows: the handover

?? is planned, but can fall back to an unplanned handover. Only proactive operation allows ensuring that the MN's service requirements can be fulfilled during and after a handover. This can be achieved by contracting with one or more candidate access routers. Also, packet loss can be avoided by buffering and bi-casting techniques even if link-layer connection set up is slow. A planned handover may not be possible or it may fail, for example, due to sudden loss of connectivity to OAR. Therefore, a graceful transition to an unplanned handover phase must be available.
?? is "mobile controlled" (i.e., the network may assist in the handover or constrain it but MN has the final control of the handover target). This is necessary because only the MN user and application may be aware of their transitory needs.
?? does not assume any special support from the link-layer (e.g., make-before-break connection) but can make use of special features. For example, make-before-break is possible if the MN supports several access technologies. Furthermore, the link layer may be able to give indications of handover-related events, which should be used to expedite handover initiation and execution.
?? assumes a "semi-static" co-located CoA[22] for the MN within an administrative domain. This means that the MN's routable IP address does not change at an intra-domain handover, which is a direct consequence of BRAIN's design principles.

[21] More precisely, also handovers between interfaces of an AR but for simplicity we mainly discuss inter-AR handovers here. A handover between access points may happen 1) within an AR but without changing AR's interface, 2) within AR but changing AR's interface, and 3) between ARs. Case 1) is not considered here because it can be solved by link-layer handover mechanisms. Case2) must be considered here, which means that when we are talking about CARs we actually mean interfaces of CARs. Therefore, the interesting level of granularity in points of attachment is an interface – not a node.

Thus, we will not assume that the MN can communicate simultaneously with OAR and CARs. However, we make the assumption that the MN is able to listen to CARs' broadcast advertisements while still being connected to OAR. That is, we don't require multihomed MNs but we assume that the MN can temporarily tune its receiver to neighbouring channels.

A3.1.5.2 Detailed Design Issues for a Planned Handover

When planned handovers are assumed and the MN is given some control over the handover, from routing perspective the essential functional issues are:

?? How does OAR or the MN know that a handover is needed?
?? How does the MN and/or OAR determine CARs?
?? How do the MN and OAR inform each other about CARs?
?? How does the MN inform OAR where it wants to handover?
?? How is MN's identification and other context conveyed to CARs?
?? How does OAR or a CAR inform the MN that that the handover is possible or has succeeded?
?? How are IP packets forwarded from OAR during a handover (to avoid packet loss)?
?? How are handover (and path update) messages authenticated?
?? How does the MN acquire new IP addresses (if any) and how is the uniqueness verified?
?? How is the handover protocol integrated with path updating?

How does OAR or the MN know that a handover is needed?

Traditional network-layer movement detection methods that rely on periodic router advertisements or monitoring forwarding progress of packet transmission may not be feasible for fast handovers.

The MN may become aware of neighbouring radio cells when listening to their beacons at idle periods of packet transmission, or the MN may be equipped with radio receivers for different technologies which allows continuous monitoring of alternative transmitters.

Link-layer triggers at either the MN or at OAR have been suggested for achieving fast handovers. In unplanned handovers the trigger may happen at NAR. Triggers may be the result of MN's link quality monitoring and/or access network's awareness of alternative radio cells, possibly of different technologies. Because triggers only involve link-layer signalling or they are local to the MN or the access network, they will not be further specified here.

Event notifications that can trigger handovers are described in the BRAIN IP_2W documentation.

How do the MN and OAR determine CARs and how they inform each other about them?

Although the MN makes the final decision on a handover, OAR should be able to suggest CARs to the MN. If OAR advertises a CAR, it should ensure that the advertised CAR is able to fulfil the MN's service requirements. This would entail a contract bidding protocol between OAR and CARs, which is not specified here. Alternatively or in addition, the MN could select a set of CARs by listening to their advertisements.

Consequently, a message is needed from OAR to the MN which advises the MN of candidates. The MN should be able to solicit this advice, and the MN should also have other means for candidate selection. Otherwise an operator could capture the MN in the domain of its own access routers.

The message from OAR should identify the viable CARs and the means for accessing them. The latter may be link-layer information that can be opaque to the network-layer. At one extreme it could contain a program for the software radio at the MN. The idea is that the MN would feed this information to its wireless device interfaces (using IP_2W control primitives). If the technology supports handover target selection by similar link-layer messaging between the MN and OAR, this network-layer advertising is not needed.

When the MN is capable of determining the CARs and the access method, the MN may request proactive preparations from the OAR for a handover to one or more CARs.

In the CAR selection process, the set of CARs converges as follows, for example:

1) At first, all OAR's "neighbouring" ARs can be viable candidates

[22] We use the term CoA to denote an IP address that is globally routable. Although this term originates from Mobile IP, the use of this term does not imply that Mobile IP must be used.

2) OAR negotiates with the neighbouring CARs based on its knowledge of the MN's requirements and based on the access network's policy
3) MN selects a set of CARs from the union of the set of CARs advertised by OAR and the set of CARs that the MN has determined by itself.
4) OAR accepts only a subset of the MN-proposed CARs as its handover peers.

Obviously, it would be beneficial if the CAR selection process converges to a single NAR at the first possible instance.

How is the MN's identification and other context conveyed to CARs?

CARs need to know the MN's IP address, link-layer address, and credentials before the MN registers with a CAR in order to be able to authorise the MN and to send a confirmation of a successful handover. This information could be sent from OAR to CARs in a message that is used for conveying other context information (like QoS, or header compression state). The context to be conveyed may include both link-layer and network-layer protocol state. Therefore, IP2W has to specify mechanisms for retrieving and setting up the MN's context at access routers. Context features and transfer issues are discussed in [A3.4] and in [A3.18].

CARs should acknowledge the handover request from OAR. This acknowledgement may not be needed if context transfer is preceded by a context/contract negotiation procedure between the access routers. If CARs have to reserve resources for prospective MNs, means for releasing these tentative resource reservations should be available.

How does OAR or a CAR inform the MN that the handover is possible or has succeeded?

After OAR has built packet forwarding tunnels to CARs, or downstream traffic is diverted to NAR by other means, the CARs could advertise their availability to the MN by gratuitously sending service advertisements, which may indicate their service capability. Based on these advertisements, the MN may select one of the advertising CARs as its NAR.

However, this kind of reactive advertising might not be feasible if only a broadcast channel can be used because it cannot be synchronised with the MNs solicitations and it wastefully spends radio resources. In particular with connection-oriented links, it would be advantageous if the selection among CARs converges to a NAR as early as possible. Then it would be more feasible to send the confirmation of a successful handover while the MN is still connected to OAR.

The MN should register with the NAR using registration messages of the micro-mobility protocol.

How are IP packets forwarded from OAR during a handover?

To avoid packet loss during a handover OAR should temporarily buffer and/or bi-cast to CARs packets that are destined to the MN. This requires establishing a tunnel between OAR and CARs because, generally, OAR and CARs will not share a link.

To prevent IP packet duplication, the MN should be able to detect duplicates, and both OAR and CARs should not transmit the same packets on the wireless link. The tunnelling phase must have a short lifetime, and it should be terminated by a CAR's or a cross-over node's request.

How are handover (and path update) messages authenticated?

A method for authorising the MN's access to the network must be specified. Additionally, a method for authenticating messages between access routers should be specified. Typically a MN acquires a session key during its initial access to a network using AAA mechanisms, and the access routers in a domain share a network key, which can be used for encrypting the MN's session key and for authenticating messages between routers in the domain. A key shared between the MN and NAR can also been used for encrypting traffic on the wireless link.

How does the MN acquire new unique CoAs?

In a mobility protocol that uses semi-static (co-located) CoAs, the MN does not necessarily need to allocate a new CoA at each intra-domain handover. EMA-MIP is an example of such protocol. Nevertheless, it allocates a new CoA at each NAR for "horizontal signalling".

The goal is to avoid any new address assignments if they unnecessarily add to latency or protocol complexity. However, a handover protocol should incorporate means for efficient address acquisition in anticipation of extending the protocol to support inter-domain handovers (where the address will change anyway).

How is the handover protocol integrated with path updating?

A path update message can be sent either by the MN or an AR. If the path update is destined to a node that does not share a security association with the access network, the path update must be originated by the MN. Otherwise, an AR could perform surrogate path updates on behalf of the MN. Especially, in strictly network controlled local operation, the MN needs not be aware of local path updates.

The message sent by the MN to NAR for registering (i.e., the message "Host-Handoff Request" in the Generalized IP Handoff Framework) may or may not be the same message that is used for path updating. The NAR will relay the message to routing infrastructure of the access network. If separation of handover and path updating is striven for, different message types should be specified for registrations and path updates.

A3.1.6 Conclusions about a Generic Handover Protocol

Based on the identification of the functional requirements and detailed design issues, the handover protocol can be sketched out. We briefly describe the planned and unplanned variation of the handover protocol at a high level without going into details of the message syntax (such as whether to use ICMP or IPv6 Destination Options for signalling).

The planned handover is performed when the MN is able to make preparations for the handover while still being connected to the OAR. The basic idea is to minimise packet re-routing delay by overlapping the required wired-network signalling with the establishment of the wireless link to NAR. If the MN abruptly loses its connection to OAR, or the planned handover fails for other reasons, the MN may fall back to the unplanned handover.

Planned Intra-domain Handover

In a planned intra-domain handover, the following course of actions can be envisioned (see Figure A3-13):

0) A link-layer trigger or another indication that signifies a need for a handover occurs either at OAR or at the MN.
1) If the trigger occurs at the MN, the MN may solicit information about CARs from OAR (*CAR Solicitation*). This trigger should is optional as it may occur when the MN itself identifies a CAR. Then the MN could directly send a *Host Handover Request*.
2) OAR responds to the MN's CAR Solicitation with a *CAR Advertisement*, which contains identification of CARs and link-layer specific advice on how to reach them. Before sending the advertisement OAR may need to perform a context negotiation with ARs in order to identify the set of CARs that can be advertised to the MN. This negotiation protocol is out of the scope of the handover protocol.
3) The MN sends a list of CARs to OAR in a *Host Handover Request*. This list may be the result of MN's own radio signal measurements and/or policy decisions, and/or it may be based on OAR's advertisement of available CARs.
4) OAR requests for a handover from the CARs that are indicated in the *Host Handover Request* with an *AR Handover Request*. This request contains MN's identification, IP address, link-layer address, session keys, and other required state information (i.e. the MN's context).
5) CARs reply to OAR with an *AR Handover Reply*, which confirms or denies the request.
6) OAR confirms the availability of CARs that have acknowledged the OAR's request by sending a *Host Handover Reply* to the MN. OAR also starts tunnelling packets to CARs.
7) In connection-oriented links the CARs wait for link establishment indications for the attaching MN. CARs buffer the forwarded packets until the link is established. CAR/NAR sends a *Router Advertisement* to the MN to advertise its services. NAR is implicitly selected by link-establishment with connection-oriented link technologies. In technologies with random access links, the final NAR selection may be deferred until the MN registers with the access router.
8) The MN registers with the network by sending a *Registration Request*.
9) NAR generates a *Path Update Request* of the micro-mobility protocol (or relays the Registration Request) towards a "cross-over" router.
10) NAR receives the *Path Update Reply* from an uplink router.
11) NAR relays the *Registration Reply* to the MN.

After the downstream routing path has been diverted to NAR, packet forwarding at OAR can be terminated. This can be achieved by a timeout mechanism at OAR or an explicit signal from a "cross-over" router or NAR. This signal is not shown in the figure.

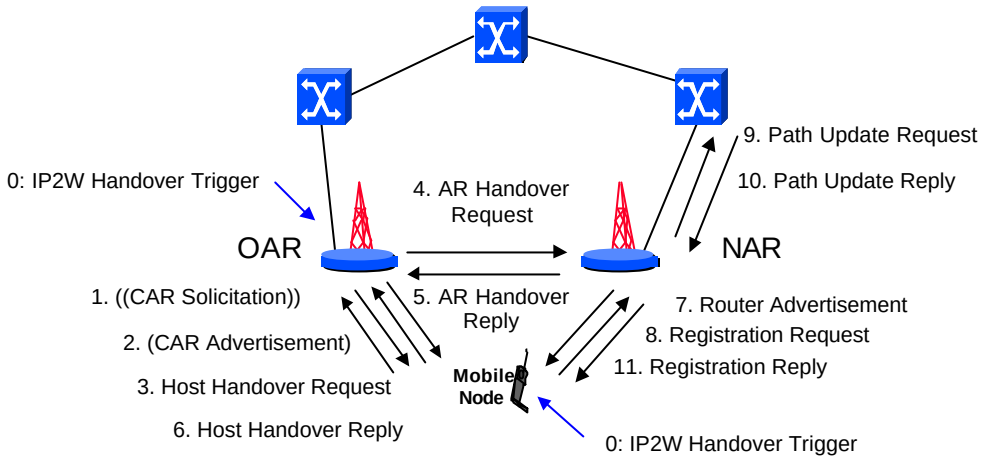


Figure A3-13: BRAIN Planned Handover

Unplanned Intra-domain Handover

In an unplanned intra-domain handover, the following course of actions can be envisioned (see Figure A3-14):

0) A link-layer trigger (e.g., loss of connection) or another indication that signifies a need for a handover occurs at the MN.
1) The MN establishes a link with NAR and, optionally, sends a *Router Solicitation*.
2) NAR responds with *a Router Advertisement*.
3) The MN registers with the network by sending a *Registration Request*. The Request includes the identification of OAR and indications of the state that should be transferred from OAR.
4) NAR requests for a handover and solicits for the MN's context information from OAR by sending an *AR Handover Request* message.
5) OAR responds to NAR with an *AR Handover Reply* message, which includes the required context information. OAR starts forwarding downstream traffic to the MN by establishing a tunnel to NAR.
6) NAR generates a *Path Update Request* of the micro-mobility protocol (or relays the Registration Request) towards a "cross-over" router.
7) NAR receives the *Path Update Reply* from an uplink router.
8) NAR relays the Path Update Reply (as *Registration Reply)* to the MN.

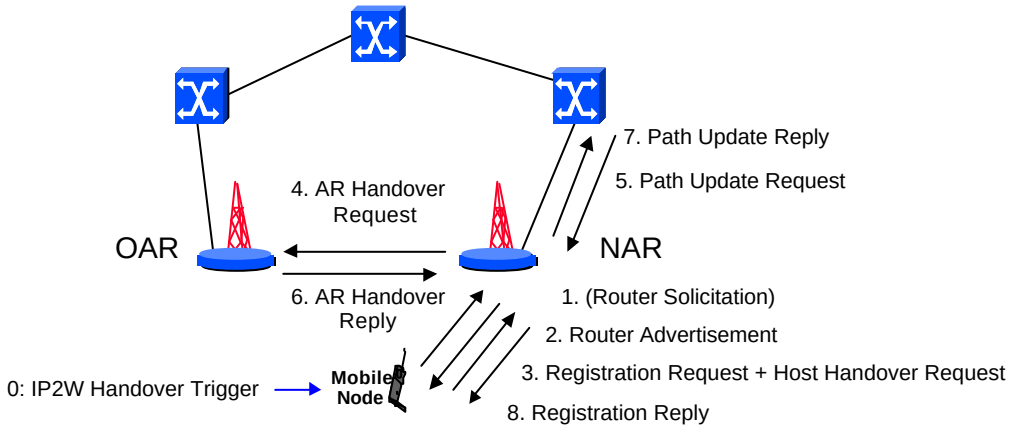


Figure A3-14: BRAIN Unplanned Handover

Note that NAR may perform path updating and consultation with OAR simultaneously if the MN's Registration Request can be authenticated without first retrieving the MN's session key from OAR.
The overall protocol is similar to "Fast Handovers for Mobile IPv6" [A3.16] proposed by the Mobile IPv6 Handoff Design Team, which on the other hand, is a clear descendant of the Generalized IP Handoff proposal [A3.2]. Our protocol complements the generalized framework by adding CAR advertisements from OAR to the MN and completing the context transfer handshake between OAR and NAR. Furthermore, in [A3.2] , the host handover reply is missing, and in [A3.13], this acknowledgement may be bi-casted through OAR and NAR.

A3.1.7 Handover Management References

[A3.1] C. Perkins and D. Johnson, “Route Optimization in Mobile IP”, Internet draft (work in progress), draft-ietf-mobileip-optim-09.txt, February 2000.

[A3.2] A. O’Neill, G. Tsirtsis, and S. Corson, “Generalized IP Handoff”, Internet draft (work in progress), draft-oneill-craps-handoff-00.txt, August 2000.

[A3.3] A. O’Neill, G. Tsirtsis, and S. Corson, “EMA Enhanced Mobile IPv6/IPv4”, Internet draft (work in progress), draft-oneill-ema-mip-00.txt, July 2000.

[A3.4] A. O’Neill, G. Tsirtsis, and S. Corson, “State transfer between Access Routers during Handoff”, Internet draft (work in progress), draft-oneill-handoff-state-00.txt, August 2000.

[A3.5] K. El-Malki and H. Soliman, “Fast Handoffs in Mobile IPv4”, Internet draft (work in progress), draft-elmalki-mobileip-fast-handoffs-03.txt, September 2000.

[A3.6] K. El-Malki and H. Soliman, “Fast Handoffs in MIPv6”, Internet draft (work in progress), draft-elmalki-handoffsv6-01.txt, November 2000.

[A3.7] R. Koodli and C.Perkins, “A Framework for Smooth Handovers with Mobile IPv6”, Internet draft (work in progress), draft-koodli-mobileip-smoothv6-01.txt, November 2000.

[A3.8] R. Koodli and C.Perkins, “A Context Transfer Framework for Seamless Mobility”, Internet draft (work in progress), draft-koodli-seamoby-ctv6-00.txt, February 2001.

[A3.9] R. Koodli and C. Perkins, “Fast Handovers in Mobile IPv6”, Internet draft (work in progress), draft-koodli-mobileip-fastv6-01.txt, October 2000.

[A3.10] G. Krishnamurthi, R. Chalmers, and C. Perkins, “Buffer Management for Smooth HandOvers in Mobile IPv6”, Internet draft (work in progress), draft-krishnamurthi-mobileip-buffer6-00.txt, July 2000.

[A3.11] G. Krishnamurthi, R. Chalmers, and C. Perkins, “Buffer Management for Smooth Handovers in IPv6”, Internet draft (work in progress), draft-govind-seamoby-buffer6-00.txt, February 2001.

[A3.12] R. Koodli, M. Tiwari, and C. Perkins, “Header Compression State Relocation in IP Mobile Networks”, Internet draft (work in progress), draft-koodli-rohc-hc-relocate-00.txt, July 2000.

[A3.13] R. Koodli, M. Tiwari, and C. Perkins, “Context Relocation for Seamless Header Compression in IP Networks”, Internet draft (work in progress), draft-koodli-seamoby-hc-relocate-00.txt, February 2001.

[A3.14] P. Calhoun, et. al., “Foreign Agent Assisted Hand-off”, Internet draft (work in progress), draft-calhoun-mobileip-proactive-fa-03.txt, November 2000.

[A3.15] C. Perkins, “Fast Handovers for Mobile IPv6”, Internet draft (work in progress), draft-ietf-mobileip-handover-00.txt, November 2000.

[A3.16] G. Tsirtsis, “Fast Handovers for Mobile IPv6”, Internet draft (work in progress), draft-designteam-fast-mipv6-01.txt, February 2001.

[A3.17] A. Campbell, et al., "Cellular IP", Internet draft (work in progress), draft-ietf-mobileip-cellularip-00.txt, January 2000.

[A3.18] O. Levkowetz, et. al., “Problem Description: Reasons For Doing Context Transfers Between Nodes in an IP Access Network”, Internet draft (work in progress), draft-ietf-seamoby-context-transfer-problem-stat-00.txt, February 2001.

[A3.19] Karim El Malki (editor), “Low latency Handoffs in Mobile IPv4”, Internet draft (work in progress), draft-ietf-mobileip-lowlatency-handoffs-v4-00.txt, February 2001.

A3.2 Paging support

A3.2.1 Basic conclusions for a paging mechanism supported in the BAN

Among all, following basic requirements were identified for the **idle mode** support in [A3.20], that will be discussed afterwards:

- should page locally to track mobile location in real-time
- should be scalable by pushing paging initiation closer to the "lower" BRAIN router[23]
- should be reliable by allowing paging initiation to occur at any router in the BAN.
- should be flexible by allowing for fixed, hierarchical or mobile-specific[24] paging areas.
- should optimise the intra-domain and inter-domain update frequency
- should minimise the intra-domain and inter-domain update latency

A3.2.2 Existing proposals

The following list of existing proposals shall not be thought of as exhaustive. It allows to identify particular paging scheme characteristics. All the described proposals only specify the idle mode support, i.e. how the stand-by is triggered is never fully described. Among all, it is possible to distinguish the ones relying on a layer 2 paging scheme (HAWAII case) and the other ones that only define a layer 3 idle mode support (Cellular IP, P-MIP, HMIPv6 cases).

A3.2.2.1 HAWAII paging scheme [A3.23],[A3.24]

In its first version, the HAWAII paging scheme relies on IPv4 and MobileIPv4.

Paging areas are statically defined using multicast IP addresses, but it should be possible to have dynamically defined paging areas (e.g. mobile-specific paging areas) as every router having multicast functionality could dynamically join a multicast group. Each paging area corresponds to a multicast group and all access routers[25], that belong to this area must join the group. The multicast IP address would generally be configured by a network administrator.

Paging entries are implemented in routing tables of the routers belonging to the path from the mobile node to the domain root router (DRR). Paging entries for a mobile node give the association between the Co-located Care-of Address of the mobile node and the multicast IP address identifying the paging area, where it is. They are created by path set-up messages sent at MN power up and handover procedures as well as by paging update messages sent by a MN to its domain root router when it detects change of paging area. They are refreshed by specific paging refresh messages, generated by each intermediate nodes between the MN and the DRR.

An access network is implicitly informed that an MN switches into the idle mode, as the routing entries are automatically erased in absence of updates. Then a packet sent to an idle MN is directly routed using paging entries. When the packet reaches a node, which contains an association for the multicast address relating to the paging area of the idle MN, the corresponding node stores the packets and sends a paging request in the paging area defined by the multicast address. These layer 3 paging request messages are not propagated over the radio link as HAWAII requires relying on layer 2 paging schemes for the air interface: they are converted by the HAWAII access routers into layer 2 paging messages. Any router, including access router, having multicast capabilities[26] can initiate the paging request. The paged MN answers by sending a paging response message to its access router which updates its routing table,

[23] here is a compromise between [small paging area and increased location updates] and [large paging area and more paging packets transmitted]. The extremities would be [one paging area per BAR] and [one paging area per domain or BAN]. Then no recommendation could be given independently from the network configuration and the mobility of the MN.

[24] User-defined is here replaced with mobile-specific, because “user” could mean the operator or the MN user; and paging areas that are defined by the operator user should be statically defined, while an MN user should not care about paging areas.

[25] the HAWAII draft specifies that all base stations have IP routing functionality. We call then these base stations "access routers"

[26] the routers that should have multicast functionality would have to be selected by the network administrator

including routing and paging entries and sends a paging reply towards the initiator of the paging request. Until the MN is in idle mode, it sends periodic paging refreshment messages to the access router which also sends paging update messages to the domain root router when it detects the MN changes paging area. In this latter case, the DRR has to positively acknowledged the paging update message, that only updates the paging entries. This detection relies on layer 2 scheme as it is done by analysing beacon signal from the access router, that has to convey the paging area identifier. This beacon signal is periodically emitted[27].

The following figure gives the state diagram of an HAWAII MN. Let's note that the "null state" is only mentioned by the HAWAII draft but will be the same state in all paging schemes.

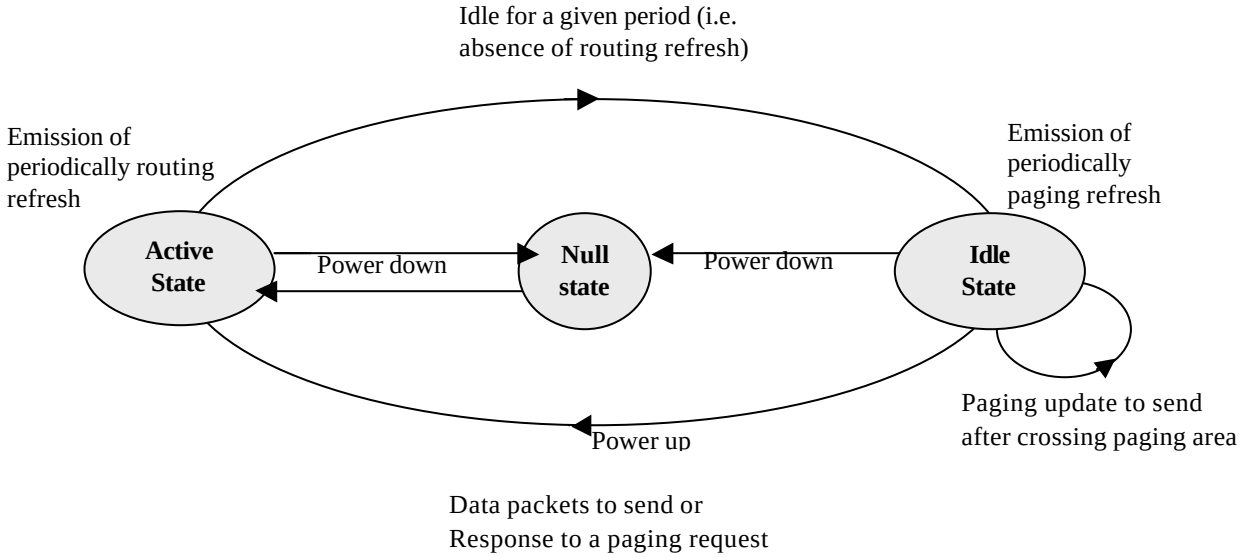


Figure A3-15: HAWAII Node State Diagram

A3.2.2.2 Cellular IP paging scheme [A3.25],[A3.24]

The Cellular IP scheme is fully handled at the IP layer as it does not require any specific support from the link level.

Paging caches that define paging areas are implemented in specific Cellular IP nodes, selected by a network operator. They are configured independently from the classical IP routing tables similarly to Cellular IP routing caches with longer lifetime. They are created and updated by route-update packets when the mobile is in active mode and by paging-update packets periodically sent or sent after a paging area change when the MH is in stand-by mode. They are refreshed by update packets as well as upstream data packets. Then a Cellular IP paging entry for an MN associates its home address with the interface of the neighbour node from which the last update packet has been sent.

A Cellular IP access network is implicitly informed that an MN switches into the idle mode, as the routing entries are automatically erased in absence of update messages. Then, when a packet sent to an idle MN reaches a node in the Cellular IP network, its paging cache is consulted. If the node does not have a paging cache, it broadcasts the packet to all its neighbours. Otherwise, the packet is either transmitted to the paging area while the node has a paging cache with an entry for the MN, or rejected if the node has a paging cache without any entry for the mobile host. When the paged MN receives the packet, it emits a route-update message that allows to create its routing entries and switch into the active mode. When an idle MN detects it changes paging area, it directly sends an ICMP paging update packet towards the Cellular IP gateway. Then specific intermediate Cellular IP nodes[28] having paging

[27] periodic fixed time slots known by the MN to monitor the beacon signal

[28] means intermediate routers in the Cellular IP network between the MN and the Cellular IP gateway

capabilities, i.e. that are configured with a paging cache, have to monitor these paging update messages to update their paging cache. Each paging area has a Paging Area Identifier, that is transmitted in periodic beacon signal by the base stations that belong to the paging area.

The state diagram of a MN is depicted in figure below.

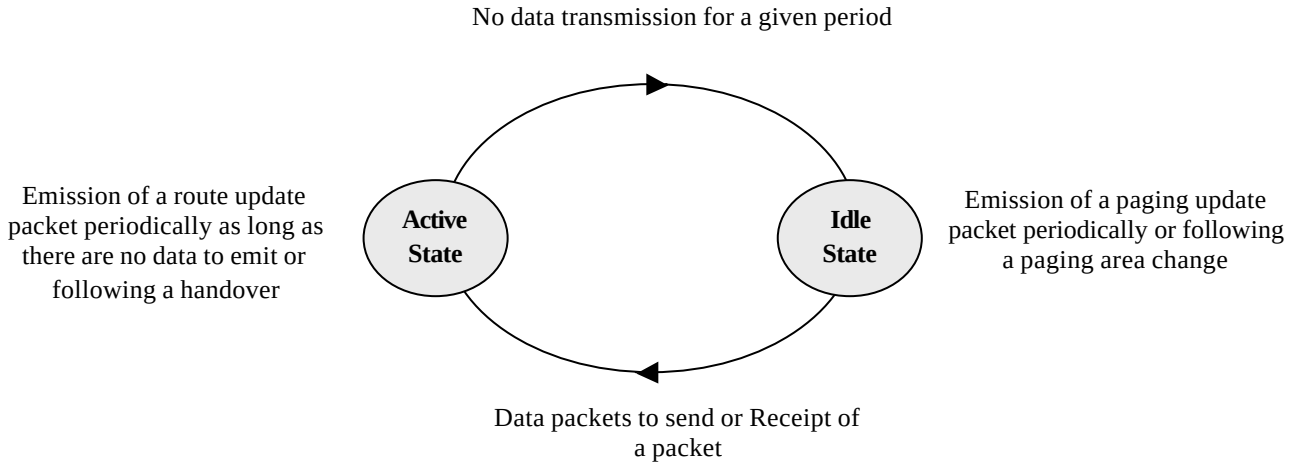


Figure A3-16: CIP Node State Diagram

A3.2.2.3 Minimal Paging extensions for Mobile IP [A3.27]

This paging scheme (P-MIP) fully relies on the Mobile IPv4 and is a pure layer 3 paging scheme.

Paging areas are defined as a group of foreign agents (i.e. generally access routers). They may be manually or automatically configured by maintaining paging tables in each foreign agent. In the case they are automatically configured, a paging server could be used allowing each foreign agent to directly acquire its Paging Area Identifier (PAI) from the server. The P-MIP supports non-overlapping paging areas as well as overlapping paging areas. A paging table of a foreign agent lists all the foreign agents that belong to its paging area. In the case of overlapping paging areas, the PAI contains all the foreign agent's IP address that belong to the corresponding paging area and is transmitted only once to the MN before it becomes idle[29], whereas in the case of non-overlapping paging areas, the PAI is just a two-octets value and is broadcast in each foreign agent advertisement. Paging Area Identifiers are directly broadcasted by the mobility agents in specific extension of their ICMP agent advertisement messages.

The paging request initiator is always the registered foreign agent, that is the Mobile IPv4 foreign agent through which an MN made its latest registration with its home agent before becoming idle. An idle MN is then registered with a care-of address related to its registered foreign agent until it performs a new registration, even if it moves and changes its foreign agent. Then an idle MN moving changes less frequently its care-of address than if it stays in the active mode.

When a packet is sent to an idle MN, its home agent forwards it towards the last registered foreign agent. This foreign agent then broadcasts a paging request, conveying the home address of the paged MN inside its own sub-network as well as towards the other foreign agents belonging to the same paging area (according to its paging table). These foreign agents broadcast themselves the paging request in their own network. When an idle MN performed a handover and receives a paging request, it has first to register with its home agent to inform it of its new care-of address before sending its paging reply towards its former registered foreign agent. When an idle MN detects it changes its paging area, it has to register with its new foreign agent.

The state diagram of a MN is depicted in figure below.

[29] through the Idle Reply sent by the registered foreign agent

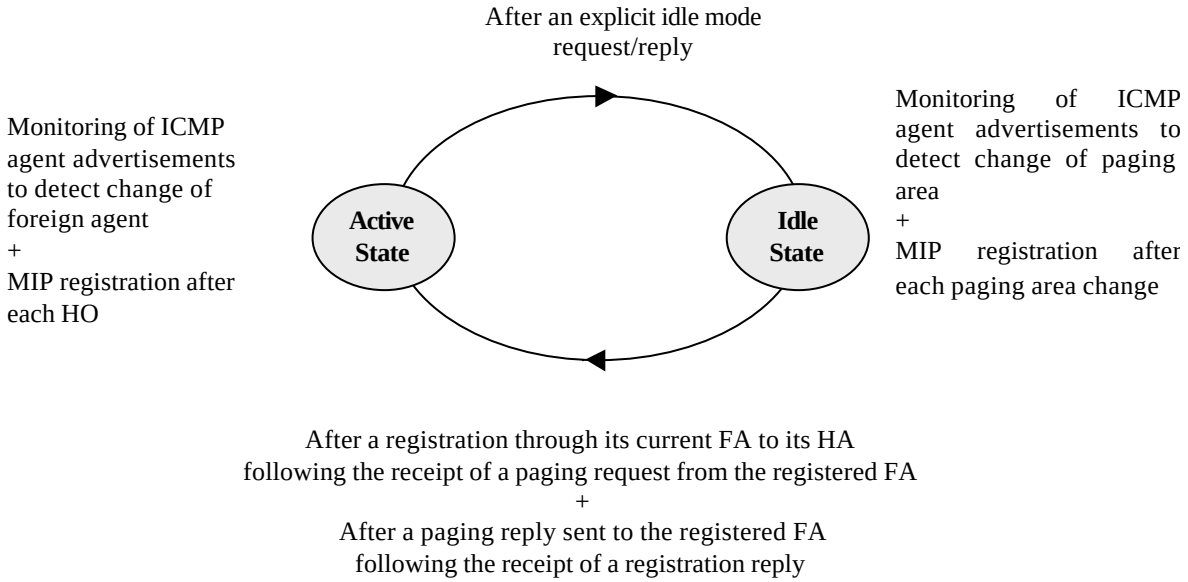


Figure A3-17: MIP Node State Diagram

A3.2.2.4 HMIPv6 paging scheme [A3.28]

The HMIPv6 paging scheme is directly deduced from the Mobile IP Regional Paging (MIRP) [A3.30], that is inspired by Mobile IPv6 extensions. Consequently, as both proposals are very similar, the current document will only go into details of the HMIPv6 proposal (one of the rare IPv6 proposal) and give the main differences compared to the Mobile IP Regional Paging.

The HMIPv6 paging mechanism relies on a central point (Paging Mobility Anchor Point[30]), that is unique per domain, to store location information concerning idle mobile nodes belonging to its domain. It represents the highest MAP (Mobility Anchor Point) in a HMIPv6 domain and may be in charge of different paging areas that could be defined at a lower level in the domain. A PMAP only is able to initiate paging request towards an idle MN and buffer data packets destined to idle MN until it receives a paging reply. The paging request consists of a classical IPv6 packet with a specific destination option[31] that can only be sent by a PMAP into a paging area, when it receives data IP packets destined to an idle MN. This destination option contains a specific field, called Paged Mobile Node Address, providing one or several regional care-of addresses[32,33] of the idle MNs, depending on the number of MNs that need to be paged. The way how a PMAP chooses the paging area in which it has to send a paging request on receipt of data packets destined to idle MN is completely implementation specific (i.e. this is not specified). When a paging request is sent into a paging area, corresponding access routers may store the received paging information and send specific Paging Router Advertisements[34] to the corresponding idle MN. The destination address of these Paging Router Advertisements is the solicited-node multicast address[35] of the idle MN obtained from its regional care-of address. In absence of paging request, the different routers below the PMAP also emit periodic Router Advertisements with a specific extension to advertise the Paging Area ID they belong to.

[30] versus a Paging Foreign Agent, that is a foreign agent at the root of the paging area in MIRP

[31] the paging request is an UDP packet in MIRP

[32] a regional care-of address of a MN refers to the MAP it is attached to

[33] in MIRP, the Paged Mobile Node Address is the IP Home Address of the idle MN, that is the target of the paging request

[34] Router Advertisements with a Paged Mobile Node Address extension

[35] in MIRP, a paging multicast address directly given by the paging foreign agent to an MN before becoming idle is used

A paging area is addressed with an IPv6 multicast address that is obtained from the Paging Area ID, permanently assigned and of global scope.

An MN informs the domain nodes that it will switch into its idle mode by explicitly sending to the PMAP an Idle Mode Request message, that the PMAP itself has to acknowledge before the information is taken into account by intermediate routers between the PMAP and the MN. In this way, the PMAP updates its binding caches to localise the idle MN (paging area ID and regional care-of address of the idle MN[36]) and the intermediate routers do not maintain any routing information for the idle MN in their routing tables.

During a HO, an idle MN keeps its regional care-of address, and consequently its solicited-node multicast address even if it changes its MAP, while it stays in the same paging area. In the case it detects a new paging area it has to perform a new idle mode registration in order to inform the PMAP of its new regional care-of address and its new paging area.

An idle MN has to periodically monitor Router Advertisements to be able to detect a paging request or a change of paging area. This requirement implies for the MN to maintain a continuous IP connectivity. In order to reduce the power-on time of idle MN, a fixed paging time slot could be optionally agreed between a domain and a MN before it switches into its idle mode. This option is a way to synchronise all access routers belonging to the same paging area for the emission of router advertisements and to trigger the stand-by mode of idle MN. How the paging time slot is agreed is not currently specified and may be implementation specific.

It should be noted that the HMIPv6RP paging mechanism strongly relies on solicited-node multicast address, that is an IPv6 specificity only ; also it completely relies on Mobile IPv6 messages and is a purely L3 paging mechanism, as it does not rely on any L2 paging functionality. Nevertheless, the interworking with a standby mode would be facilitated using the optional agreement of the fixed paging time slot.

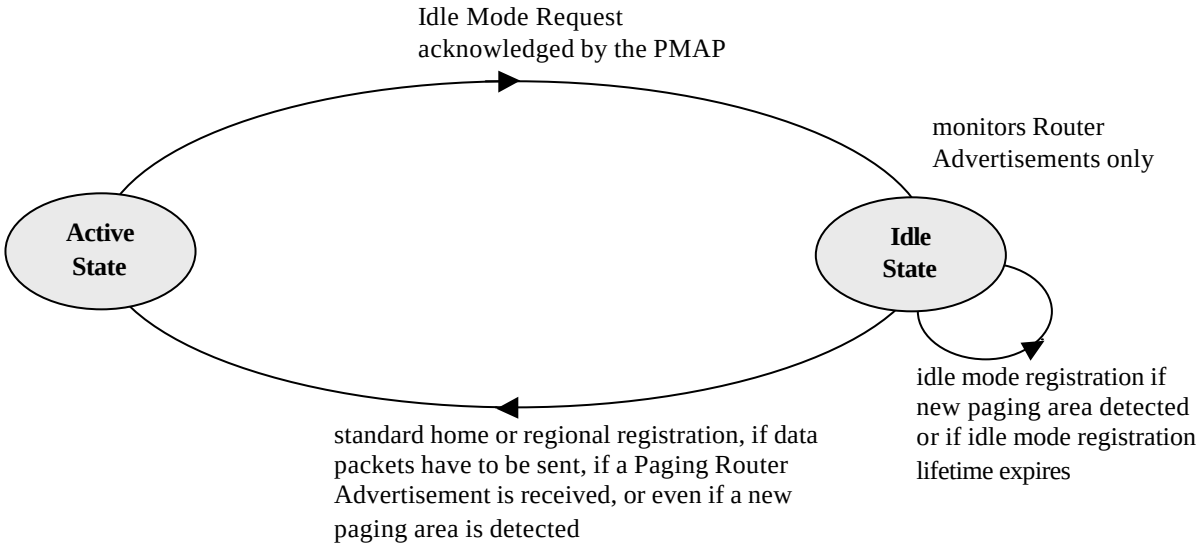


Figure A3-18: HMIP Node State Diagram

A3.2.2.5 Castelluccia paging scheme [A3.29]

This specification only defines a way to have dynamically mobile-specific paging areas. It should be thought of as a particular characteristic of the paging scheme rather than a complete paging scheme proposal. It defines how the paging scheme could support adaptive individual paging areas, allowing each MN to have its own paging area. This proposal implies that all MN have to run a specific algorithm to determine their paging areas.

[36] the current draft does not specify how a PMAP decides to use a common multicast address or not to page several MNs at the same time

A3.2.2.6 Comparison of paging scheme characteristics

	HAWAII	Cellular IP	P-MIP	HMIPv6RP
Mobile IP version	MIPv4	MIPv4	MIPv4	IPv6 and MIPv6
Paging scheme type	L3 relying on L2 paging process	fully L3	fully L3	fully L3
Paging area configuration	- using static or dynamic multicast group address configured by a network admin - one PAI[37] per area transmitted in periodic broadcast channel by AR	- using static paging caches configured by network admin - one PAI per area transmitted in periodic broadcast channel by AR	- using paging tables given a list of foreign agents - non-overlapping or overlapping paging areas - one PAI per area transmitted in ICMP agent advertisement msg	- static IPv6 multicast address obtained from the PAI - one PAI per area transmitted in router advertisement msg
Paging information location	partially distributed in selected paging able routers having multicast capability, through paging and multicast entries in routing tables	in specific Cellular IP nodes having paging caches	- paging tables configured in each foreign agent - optionally use of a paging server	centralised in a PMAP (unique per domain)
Paging request initiator	every selected paging able router	no paging request sent	registered FA	PMAP
How to inform the network of the idle mode switching ?	implicitly with the absence of refresh messages	implicitly with the absence of update messages or upstream data packets	explicit Idle Mode Request/Reply	explicit Idle Mode Request/Reply
How to inform the access network of an HO crossing a PA in idle mode ?	paging update msg sent by the nAR towards the DRR creating new paging entries in all intermediate routers	IP paging update msg sent by the idle MN towards the Cellular IP gateway	registration of the MN with the new foreign agent	
Interworking with stand-by mode support	not specified but relies on beacon signal monitoring	not defined	not defined	optional agreed time slot to emit Paging Agent Advertisement
MN requirements	MIPv4 compliant	MIPv4 compliant - emission of specific ICMP control packets to update paging caches	MIPv4 compliant	MIPv6 compliant specific IPv6 destination option and Binding destination option
AR requirements	- multicast capability - emission of specific refresh,			several specific extension for router advertisements

[37] Paging Area Identifier

	update paging msg			
destination address of paging request	home address or CCOA directly acquired by an MN in a foreign domain when active	home address of the idle MN	home address of the idle MN	solicited-node multicast address

A3.2.3 Recommendations for a BRAIN paging scheme

According to the study of the existing proposals, a list of paging properties, that are more or less independent from the MM protocol, is drawn up. Each of these properties is discussed and recommendations are given to design a BRAIN paging scheme. The different properties may have different importance levels.

A3.2.3.1 Support of standby mode and idle mode

According to the identified requirements for the BRAIN access network [A3.20], it clearly appears that the BRAIN mobility protocol has to completely specify a paging mechanism including:

- on the one hand, the support of the stand-by mode to limit power consumption in a MN,
- and on the other hand, the support of the idle mode to minimise the radio spectrum use and the routing update messages in the BAN, when no IP data packets have to be sent to/from a mobile node.

These two aspects are completely different and they may be separated into two independent notions, let's call **stand-by mode** and **idle mode** respectively.

The **stand-by mode** is a completely layer 2 function. Its main goal is to save battery by allowing the MN to turn off its radio for a given time (of course during this sleep period it is completely unreachable), and after a wake up (e.g. pre-configured with a periodical wake up time), the MN is able to monitor one broadcast channel in order to find out whether there is pending data or at least a paging request sent to it or it changes its paging area. Then it may either become active or continue sleeping. (mainly HIPERLAN/2 approach [A3.21]). This layer-2 functionality is only relative to the wireless link technology (e.g. between a BAR and an MN) and is out of scope of the current document.

The **idle mode** is a layer-3 function, that could also rely on layer 2 functionality on the wireless link (e.g. to forward paging request between a BAR and an MN [A3.22]). Its main goal is to reduce location update signalling in the BAN. It could also allow to speed up routing by maintaining two tables: one table, that would probably keep a small size, contains entries relating to active MNs (i.e. MNs able to exchange IP packets with correspondent nodes), and a second table, that would be obviously bigger, contains most of the MNs that are currently not maintaining IP traffic but still reachable. This last particularity will depend on the BRAIN MM protocol design, whether it relies on host routing or not. Cellular IP specifies an example of idle mode that is a purely layer 3 idle mode in the sense that an idle MN has to directly monitor IP packets to determine whether it has to switch into the active mode or not. On the contrary, HAWAII relies on a layer 2 idle mode capability, as an idle MN has only to monitor a specific layer 2 broadcast paging channel to identify a paging request. It was concluded in [A3.20] to focus on a solution combining paging requests sent through link layer signalling between the BAN and the MN and location updates sent through network layer signalling. This choice, made at the expense of an universal paging system only relying on IP layer functionality, appears optimal to minimise overhead in the BAN and radio spectrum use[38].

It is the role of the IP_2W to enable the interworking between the stand-by mode and the idle mode, e.g. to trigger the standby mode when an MN is idle. The IP_2W may indicate:

- to the layer 3 that the radio systems has stand-by mode capabilities (layer 2 functions),
- to the layer 2 that the MN switched into its idle mode (i.e. MN is inactive in IP sense) to trigger the stand-by mode,

When in stand-by mode, the MN then periodically wakes up to monitor a layer 2 broadcast paging channel, that will mainly consist of beacon signals sent by the access routers conveying for instance paging area identifier and paging request. It is then required that periodic fixed-time slots are agreed between all BARs and an MN before the MN becomes idle. And when an idle MN detects it does not

[38] This choice is a prerequisite for the support of a stand-by mode (or at least it facilitates the support of the stand-by mode)

receive any more the paging channel it will have to wake up to be able to synchronise its radio transceiver with the periodic paging fixed-time slots. It is out of the scope of this document to define the IP_2W capabilities as they will be directly defined in the IP_2W working document, i.e. broadcast of paging area identifiers, forwarding of paging request on layer-2...

A3.2.3.2 Paging area configuration

The paging areas could be configured either by an operator or individually by the mobile host itself as it is suggested in the Castelluccia's proposal [A3.29], for example. In order to keep the simplicity of the paging scheme and because the added complexity of adaptive individual paging areas may be not well justified, it is recommended here to let the BRAIN network operator define the different paging areas in its BRAIN network. That means that the paging areas are more or less statically defined, firstly configured at the initialisation of the network and eventually reconfigured if the configuration of the network changes.

A paging area is defined as a set of BARs and the use of an unique multicast IP address per paging area appears to be the easiest way to identify different paging areas. Two paging areas can eventually[39] overlap in order to reduce excessive signalling in the BAN when idle MNs frequently move between two paging areas. Using multicast IP address, each BAR belonging to a paging area has to join the multicast group. Nevertheless if the IP multicast capability is not supported by each BAR, an equivalent fashion could eventually replace the use of multicast IP addresses to identify the different paging areas in a BAN. E.g. each BAR could maintain a paging table containing the IP address of all BARs belonging to its paging area. Also to facilitate the configuration of the different paging areas, even if this leads to a single point of failure problem, a central paging server, that would be updated by the network operator each time the paging area configuration has to be changed, could be used to automatically diffuse paging information in the BAN.

A3.2.3.3 Paging information update/update

How the paging entries are defined, set up, updated and refreshed directly depends on the MM protocol, since a paging entry will replace the regular routing entry for an idle MN. We consider here that a paging entry has the following generic configuration: *P: identifier of the idle MN -> multicast IP @ of its paging area*. The number of BRAIN nodes that own this kind of paging entries depends also on the MM protocol and on the network entity that has to initiate the paging requests.

Concerning the way to inform the BAN that an MN switches into the idle mode, two solutions are envisaged:

- either the BAN is explicitly informed, since messages like Idle Mode Request/Reply are exchanged between an MN and its BAR, as soon as the MN detects it can switch into the idle mode.
- or the BAN is implicitly informed, maintaining timer in BAN nodes and in the MN ; when the timer expires in absence of upstream data from the MN, the BAN deduces that the MN becomes idle.

We would suggest here to retain the first solution (explicit messages), since it would facilitate the synchronization of the paging information relating to an MN kept in the BAN and the states of the MN (active, idle or off). Also it would be possible to take into account other events apart from timer expiration to trigger the idle mode of an MN, like for instance the absence of TCP connection. However in both cases, the event that triggers the idle mode of an MN will also trigger the emission from the MN of the message responsible for the set-up of the corresponding paging entries in the BAN. Also this message could be used to trigger the storage of the MN context in the current Access Router, that could be transferred to a new Access Router when the MN switches back to the active mode. The message responsible for the update of the paging entries mainly depends on the MM protocol. For example, in the BRAIN Candidate Mobility Protocol case (cf.A3.2.4) :

- if the MN detects that it enters for the first time in a new paging area it may just perform a regular unplanned Handover (HOFF message) between its old Access Router belonging to the old paging area and the actual Access Router which is belonging to the new paging area. It means that the MN temporary enters active mode to perform the Handover and it has to re-enter idle mode at its actual Access Router.

[39] this would be an option of the paging scheme

- if the MN detects it changes its paging area and gets too far from its old Anchor then it first performs the above mentioned unplanned Handover to enter the new paging area. Afterwards the MN may change Anchor by the regular LOGIN procedure (LREQ message).

Concerning the way how the paging entries are maintained in the BAN and the way how the BAN is informed that the idle MN is still reachable (i.e. it does not switch off), two alternatives are possible :

- either the paging entries are explicitly refreshed by paging refresh messages directly sent by the idle MN itself ; in absence of refresh message the BAN deduces that the idle MN leaves the network ; this process requires use of timers in both BAN and MN,
- or the BAN keeps the paging information unchanged while the idle MN does not send an explicit LOGOUT message (depends on the MM protocol) to inform the BAN that it leaves the current paging area. In that case, a timer with a long period should be maintained in the network as well as in the MN, that would indicate when it expires that the MN leaves the network without LOGOUT (e.g. MN fails or is out of range).

We would suggest here for simplicity to reuse an explicit LOGOUT message defined in the MM protocol.

A3.2.3.4 Paging request management

The paging mechanism could alternatively imply :

- either an explicit paging request ; in that case, when data packets sent to an idle MN reach the BAN, these packets are first buffered and a explicit paging request is multicasted in the paging area to search the idle MN ; this request requires a reply from the MN to inform the BAN that it switches back to the active mode and to allow the data packets to be forwarded towards the MN,
- or an implicit paging request in the sense that the data packets are directly multicast in the paging area, i.e. they are directly used to page themselves the idle addressee MN.

According to conclusions drawn in the BRAIN deliverable D2.1, the explicit paging scheme appears generally as a more efficient scheme, although an implicit paging scheme would facilitate the support of real time services. E.g. an implicit paging request would not easily allow an idle MN to trigger its standby mode, as the idle MN should still be able to receive IP data packets. On the contrary, the explicit paging request could be conveyed in a specific broadcast paging channel that a standby MN would have to periodically monitor. Also in some cases, the diffusion of data packets to page an idle MN may overload the BAN. Let's note that this last drawback of an implicit paging request is less significant if it is assumed that an idle MN cannot receive burst of downstream packets that are not preceded by a specific short data packet exchange between the correspondent node and the MN.

Considering that an explicit paging request has to be sent to page an idle MN, we would recommend the following process:

- data packets sent to an idle MN firstly reach the last BAR, that registered the MN when it was active,
- then the corresponding BAR buffers the data packets and initiates a paging request towards all the BAR belonging to the same paging area ; all the BAR, including the paging initiator BAR, broadcast this paging request in their own sub-network in a layer 2 broadcast paging channel.

This proposal is an alternative solution to the one that implies a central paging node (for example the root node of a paging area), that would be responsible for a whole paging area and would be the only node that would be able to send paging request in the corresponding paging area. Then this proposal allow to quickly reach an MN in the case it frequently switches into the idle mode without moving during a established connection, since data packets directly reach the relevant BAR. Also it avoids on the one hand a single node of failure relating to the paging functionality and on the other hand a non-optimal routing of data packets sent from neighbour corresponding node belonging to the same paging area: in every case, these data packets would have to firstly reach a central paging node before the paging request is triggered.

Since in the proposed mechanism the BARs are the only entities that are able to send an explicit paging request, all the BARs must be initially configured to have an entry for the multicast IP address relating to the paging area they belong to. The way how they are configured depends on the paging area configuration (see §A3.2.3.2). Also paging entries relating to idle MN (described before in §A3.2.3.3) will only be owned by the BAR that serves them, i.e. none of the other routers in the BAN will have paging entries.

A3.2.3.5 Idle MN identifier

During the discussions relating to this topic, i.e. how an idle MN should be identified in a paging request, two kind of suggestions were given:

- either an IP address should be used, as it is performed for active MN ; this IP address depends on the MM protocol ; at least it should be the address the MN acquired when it was in active mode and should be static while the MN stays in the same paging area,
- or another kind of identifier like for example an NAI (Network Access Identifier), that would allow to release an IP address when an MN becomes idle.

We would recommend here to retain the first proposal, as it is more in accordance with the BRAIN design principles (e.g. layer transparency). Moreover the advantage of releasing an IP address is valid only if the IP address can be reallocated to another MN, which is not always the case. However this issue should not have a strong impact on the whole paging scheme, since it could just be thought of as a parameter of the paging request.

A3.2.4 Paging scheme with the BRAIN Candidate Mobility Protocol

The following scheme describes how the paging scheme previously retained could be integrated with the BRAIN Candidate Mobility Protocol, what could define the paging solution for BRAIN. Two scenarios are given:

- first scenario : the MN becomes idle, moves in the same paging area and receives a paging request
- second scenario : the MN becomes idle, changes its paging area and decides to change its Anchor, and switches off.

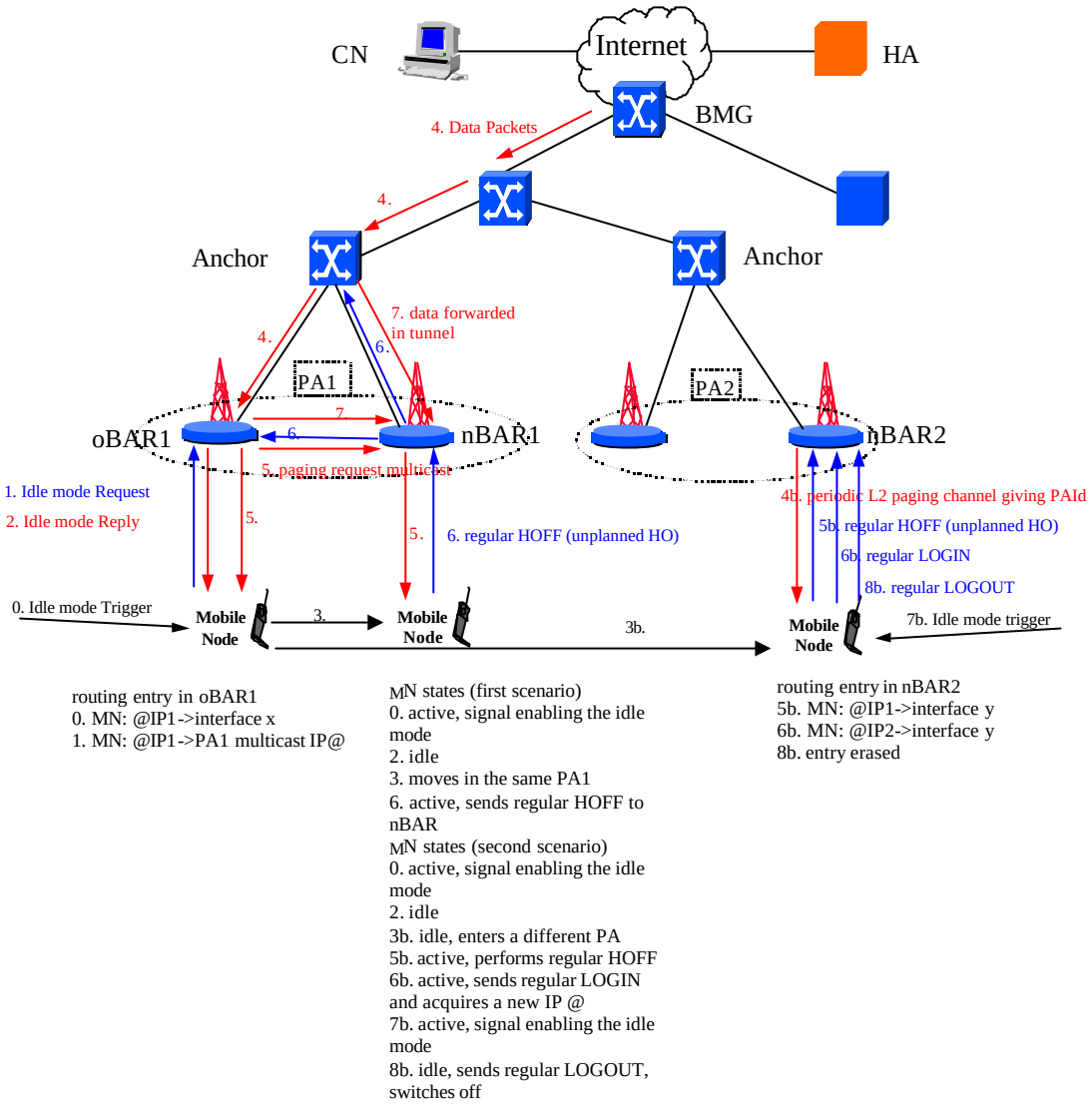


Figure A3-19: Paging with BCMP

Remarks :

- concerning message 6, that should be thought of as a response to the Paging Request at L3 (the Paging Reply is part of the BCMP protocol, during the regular unplanned Handover, a regular HOFF is forwarded towards the oBAR1 and the Anchor to install a tunnel between the oBAR1 and nBAR1 and between the Anchor and nBAR1 to start the bi-casting of data packets until the route to reach the addressee MN is not fully updated.
- if the MN changes its paging area without changing Anchor, a regular HOFF message without handover preparation (regular unplanned handover) should be sufficient.
- if the MN changes its paging area and detects another Anchor, that is optimal, it should first send a regular HOFF and then send a LOGIN message towards the new Anchor to acquire a new IP address;
- an idle or active MN should send a regular LOGOUT message to switch into the off mode.

A3.2.5 Paging References

[A3.20] Nikos Georganopoulos, Ian Groves, "BRAIN Access Network requirements, specifications and evaluation of current architectures and technologies and their requirements: core network and air interface" IST-1999-10050 BRAIN D2.1

[A3.21] Bonjour S., Bertin P., Hischke S., Kadelka A., Lott M., West M. "IP convergence layer for HIPERLAN/2", BRAIN Workshop London1 20th November 2000.

[A3.22] Laukkanen A., Bertin P. Corairie S. Liljeberg M. Suihko T. "IP to Wireless Convergence Interface", BRAIN Workshop London1 20th November 2000.

[A3.23] Ramjee R., La Porta T., Bell Labs, Lucent Technologies, "Paging support for IP mobility using HAWAII" (Internet Draft).

[A3.24] Guillouard K., Bertin P. Khouaja Y., "Evaluation of Per Host Forwarding protocols", IST-1999-10050/FT/WP2/PI/I/005/b1.

[A3.25] Campbell A., Gomez J., Wan GY, Kim S., Columbia University, New York. Z. Turanyi Z., Valko A., Ericsson Research, "Cellular IP" (Internet Draft).

[A3.26] Guillouard K. Bertin P. Khouaja Y., "Completion of Framework Evaluation - final comparison of Per Host Forwarding protocols", IST-1999-10050/FT/WP2/PI/I/006/a1

[A3.27] Zhang X., Castellanos J., Campbell A., Sawada K., Barry M., "P-MIP: Minimal Paging Extensions for Mobile IP", draft-zhang-pmip-00.txt, July 2000 (Internet Draft).

[A3.28] Sarikaya B., Haverinen H., Malinen J.T., Magret V., "Mobile IPv6 Regional Paging", Nov. 2000, http://search.ietf.org/internet-drafts/draft-sarikaya-mobileip-hmipv6rp-00.txt

[A3.29] Castelluccia C. "Extending Mobile IP with adaptative individual paging : a performance analysis"

[A3.30] Havarinen H., Malinen J., "Mobile IP Regional Paging", draft-haverinen-mobileip-reg-paging-00.txt, June 2000 (Internet Draft).

A3.3 Path Updates

This annex contains a summary of the design issues for path updates within a BRAIN Access Network (BAN). This includes issues that affect the way that path updates operate and interactions with other design issues within the BAN.

A3.3.1 General

There are two routing problems to solve within the BAN. The first of these is the fixed routing problem of traditional IP networks. Nodes within the BAN infrastructure will need to communicate between themselves. They can do this in exactly the same way as a they would in a standard, fixed network. The other, more complex case manages routing to mobile nodes (MNs). Packets need to be routable, via a BRAIN Access Router (BAR), to an MN and as this point of attachment can change, the network needs to track the location of the mobile.

This is also broadly a description of the packet forwarding design issue. The role of the path update mechanism is to install host-specific state within the BAN to allow the packet forwarding to adapt to the location of the MN.

A3.3.1.1 Routing to mobile nodes

In a fixed network, IP addresses are allocated and aggregated for efficient routing. The addresses tend to have an implicit geographical significance. For example, from the core of a network, an aggregate of networks will be reachable via a particular link. As a packet is routed closer to its destination, the addresses become increasingly specific until the final subnet is reached.

Allowing a node to move (whilst maintaining the same IP address) weakens the implied geographical significance. So conceptually, for mobile routing, there is a need to maintain location and routing information for each node.

A3.3.2 Topologies

This section discusses the impact that certain topological features may have upon the path update mechanism. It is worth noting at this point that it is considered desirable for a BRAIN path update scheme to be independent of network topology. Whilst the problem may be simplified by choosing a particular topology, this may not be true for other components of the mobility architecture and does not help the migration of legacy networks.

A3.3.2.1 Hierarchical

A strictly hierarchical case is the easiest case to consider. Upstream routing is simple, and there is only one downstream path. Changes to this path always occur at a well-defined cross-over router. This can be clearly seen in the diagram below (Figure A3-20) – as the MN moves from the oBAR (the old BAR) to the nBAR (new BAR) the path changes locally at the cross-over router (XOR). It is equally clear that a network built on a single, strict tree has serious robustness issues.

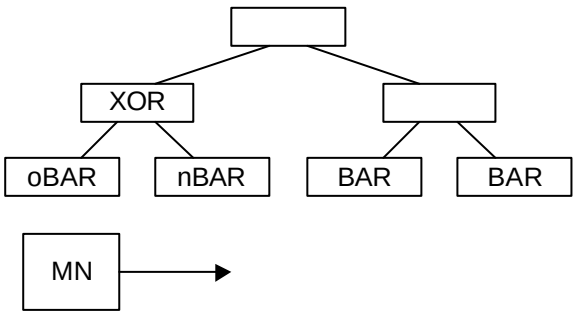


Figure A3-20: Hierarchical Path Updates

A3.3.2.2 Mesh

In a fully or partially meshed network there is more choice and, therefore, more complexity. The implications of this are considered in more detail in later sections. However, most obviously, as can be seen in the diagram (Figure A3-21), with multiple paths between nodes, the path update mechanism needs to be selective about what path information is installed. (Note, for example, that there need not be a well-defined cross-over router in this case).

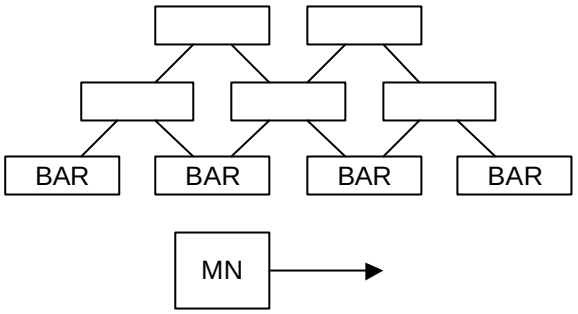


Figure A3-21: Meshed Path Updates

A3.3.2.3 Support for multiple BMGs

On the assumption that it is desirable to support multiple BRAIN Mobility Gateways (for a number of reasons) [A3.31], then there may be a need to signal path changes to one or more BMGs. However, since this possibly depends on decisions regarding the nature of routing and signalling between the BMGs, it is not being considered at the moment.

It is also noted that if multiple BMGs are being supported then there is a need both to support and control the advertisement of routes outside of the BAN. It is not clear where this functionality should be considered.

A3.3.3 Issues

A3.3.3.1 Address Aggregation

In general, address aggregation is not a direct concern of path updates. Aggregation, by allowing routing to networks rather than hosts, reduces the information stored in the network. This is enabled, to a first approximation, by the address allocation strategy and the nature of the packet forwarding method. Consider the two alternative approaches: a per-host-forwarding, hop-by-hop scheme and a tunnel-based, gateway-centric scheme.

In the per-host-forwarding scheme the fixed and mobile routing problem are typically combined into one protocol. Thus for the address of the MN to be aggregatable, it must be allocated based on the BAR to which the MN attaches. With this model, it is assumed that BARs have a pool of 'locally owned' addresses. (Note that in the degenerate case, more commonly considered with ad-hoc networking, BARs own no addresses and every route is per-host and installed by path updates).

With the gateway-centric method, the location of each MN will be tracked by at least one gateway. The fixed and mobile routing are decoupled, so the tunnel runs across the fixed network and takes advantage of normal fixed network aggregation. Aggregation for the mobile node addresses is achieved by having the gateways owning pools of addresses. This allows a gateway to only have to advertise a single route to the block of addresses that it owns.

In any case, a path update will install per-host state in the network whenever a mobile changes point of attachment without changing address. Where address aggregation is being used, there may be a need to control the scope of the path update. Specifically, to control how far 'into' the BAN host routes should be pushed. (This is closely related to the issue of edge-mobility, which is covered in the next section).

Additionally, it is assumed that if address aggregation is being used, there will be a mechanism for an MN to return an IP address to the pool. When this happens, it is expected that there will be some host specific state for this address in the network. If this is soft-state, then it may be practical and simplest to let this state expire before reassigning the address. However, there may also be an argument for using a path-update message to 'poison' the obsolete path information.

A3.3.3.2 Edge Mobility

Whether or not address aggregation is assumed, the question of how far 'into' the BAN host routes should be pushed still exists. In the edge-mobility view, path updates tend to be localised, re-routing packets around the edge of the BAN. This avoids pushing host routes too far into the network, but may result in sub-optimal routing. The most obvious ways of avoiding this is to reallocate addresses periodically and to 'optimise' the host routes. The first of these may require routing state to be purged from the network and is intrusive if performed too frequently. However, this can be controlled (for example by giving the MN some choice over when to return its address). The second alternative would probably involve flooding messages through a significant subset of the BAN to push per-host state further into the network. This is considered impractical, as it would have to be performed on a per-destination basis – implying a high signalling overhead and additional state to be managed. Where edge mobility is considered, it is assumed that address reallocation is used to control routing inefficiency.

A3.3.3.3 Tunnels

The use of tunnels is not significant to path updates. It is the topology of the nodes in which per-host state is installed that affects the mechanism that is used. For example, a network built on a per-host forwarding scheme with a tree topology has identical path update characteristics to a strictly hierarchical tunnelled network (compare Figure A3-20and Figure A3-22, where the clouds indicate arbitrary numbers of non-BRAIN aware routers).

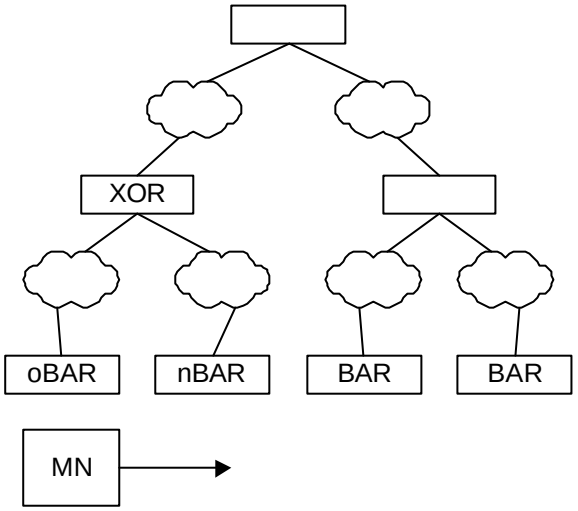


Figure A3-22: Hierarchy with Tunnels

Equally, as soon as there is any more complex connectivity between BRAIN forwarding nodes then, whether these are directly connected or linked by tunnels, the distribution of per-host state involves choices. For example, in this case it is much more likely that the dissemination of per-host state should be localised. This is shown in the diagram below (compare Figure A3-21 and Figure A3-23, again the clouds represent tunnels across arbitrary numbers of non-BRAIN aware routers).

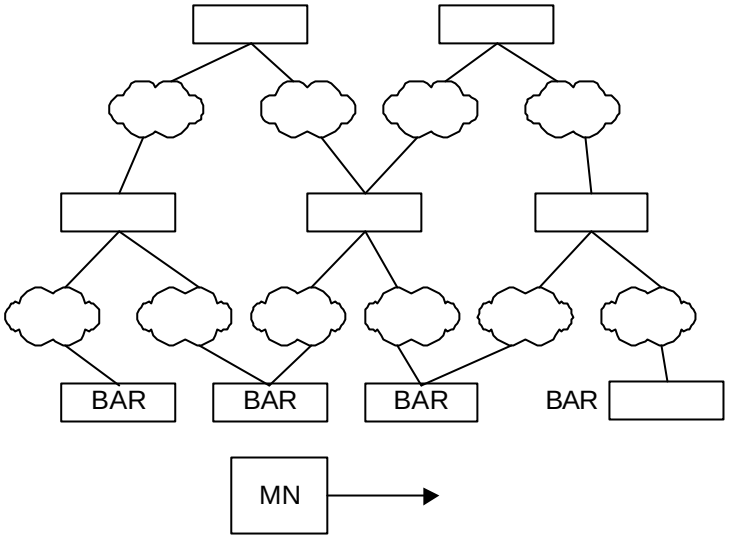


Figure A3-23: Partial Mesh with Tunnels

Fundamentally, per-host state needs to be installed in a subset of the BRAIN packet-forwarding nodes. Whether these are linked by tunnels or direct connections is unimportant.

A3.3.3.4 Source routing

There is very little difference between the use of source routing options and tunnels. They essentially achieve the same effect, one by encapsulation and one by IP options. The implication of using source routing (particularly with a DSR heritage) is that the BMG chooses *all* the pinning points of the path and, therefore, all path updates must be sent to one or more BMGs. (This may also be true of tunnels, but is less common with, e.g. HMIP). Tunnels may be considered to have a higher overhead, in terms of the encapsulation header, but are architecturally cleaner.

A3.3.3.5 Mobile to mobile routing

This appears, at first glance, to be purely an issue for packet forwarding. It is likely that this is the case, though this may depend upon the nature of the routing.

For example, if mobile to mobile routing always goes through a BMG (note that in the case of generally hierarchical topologies this does not have excessive effects on path length) then there should be no effect on path updates. If, however, mobile to mobile routing is intended to be optimal (or near optimal) within the BAN, then this may affect the installation of host routes.

Initially it is assumed that there is no necessity to design the path update mechanism in such a way as to optimise inter-mobile routing. It is worth pointing out that there may be good reasons for having mobile-to-mobile calls routed in the same way as calls to hosts in other networks (i.e. passing through a fixed point such as a BMG). These include billing and accounting, and the provision of lawful intercept facilities.

A3.3.4 Interactions

Although the design of the path update mechanism is seen as largely independent in the specific details of its operation, it cannot be considered in isolation. Therefore, this section considers a number of the possible interactions between the path-update part of a mobility solution and other aspects.

A3.3.4.1 Handover Management

It is assumed that the handover management design issue addresses the use of techniques such as temporary tunnels to ensure a smooth handover. Given this, the interaction between these design issues should be only at the level of the triggering of the handover event and requiring some signalling of the completion of the path update.

It is suggested that the solution of the handover management problem weakens any requirement for the path update to be highly localised, as the transitional period is covered by the handover. Clearly the path update must still complete rapidly enough to allow the network to converge when a MN moves. Also,

localisation of signalling may be useful for other reasons, such as achieving scalability for large numbers of MNs in a wide-area BAN.

However much the path update solution attempts to localise signalling, there will always be acts of mobility that will cause signalling to traverse the full 'depth' of the BAN. While this may not happen frequently, if it can occur it must be handled. Given this, the existence of local handover management implies no strong need for path updates to be highly localised.

A3.3.4.2 QoS

It is possible that there will be interactions between the path update mechanism and quality of service. (Generally these also affect packet forwarding; and are discussed below).If the idea of reactive path updates is considered (for example finding a downlink path for a flow when required), then this could take advantage of the requested QoS for the flow. The increase in signalling latency for setting up the question would need to be offset against the possible benefits in efficient resource utilisation.

A3.3.4.3 Packet forwarding

Packet forwarding covers the 'routing' of packets within the BAN. It is assumed that this can only make use of *a priori* routing information, or information installed by path updates. There are, therefore, interactions between these aspects, since packet forwarding would only be able to support mobility through the use of routing state from path updates.

There are a potentially a number of issues, outlined below:

Multipath Routing

It is anticipated that there will be some degree of meshing in the BAN topology. Given this, there will be multiple possible routes from an ingress BMG to a BAR/MN and *vice versa*. Fundamentally the *choice* of which path to take is down to the packet forwarding design. However, the forwarding engine can only make use of routes of which it is aware. There are two extremes for support of multiple paths.

Firstly, the path update scheme can proactively install all possible routes. Practically, this would have to be restricted somehow, to avoid every router containing state for every mobile node.

Secondly, the scheme can install any (suitable) route when a path update is required. If it becomes necessary, other routes may be reactively discovered. In this case it is not clear how the discovery process would be triggered, nor how the path update scheme would find and install the route.

QoS Routing

This is just an extension of the previous point. There may be advantages in choosing routes through the BAN based on QoS parameters. To do this, alternative routes must be known (or must be able to be discovered). Essentially the same comments made above still hold.

Resilience

This is an obvious reason to have multiple paths available. It is less clear that this requires proactive support. Perhaps the main question is how the failure condition is detected and signalled – it is reasonable to assume that a new route can be installed if initiated in a suitable manner.

Route Symmetry

Generally speaking, most of the issues to do with path updates affect installing state for downstream routing. The per-host forwarding protocols, certainly, tend to assume the use of default routes to reach an egress BMG.

In practice, route symmetry is mostly a packet forwarding issue but, since there are interactions between these two components, it is possible that there might be considerations that need to be applied to path updates.

A3.3.4.4 Security

Path updates are an obvious vulnerability – it is easy to conduct a denial of service attack by installing spurious routes in the network. Perhaps the biggest implication for any proposed solution is the effect

that such considerations would have on the use of pure data packets to refresh or introduce soft-state routing information.

A3.3.4.5 Paging

In the sense that path updates and paging are both concerned with tracking the location of the mobile, there may be some overlap. Most obviously, taking the example of Cellular IP, a message that refreshes soft-state routing information for a destination can equally refresh paging information.

If routing is hard-state, then the connection is less strong (if it exists at all). The strength of the connection would also depend upon where in the network the information was stored. (For example, if paging information is stored in a separate, central server then there is no overlap between paging and path updates).

A3.3.5 Solutions

A3.3.5.1 Attributes of Solutions

Many of these have been discussed in earlier sections, however it is helpful to collate the factors that are seen as being desirable attributes of any path update scheme.

?? Support address aggregation. As a general efficiency issue rather than as a direct function of path updates, this needs to be supported.

?? Be independent of network topology. It is understood that BANs may be expected to scale over a wide range. There may also be the possibility of migrating legacy networks. In both cases it is obvious that ideally the path update mechanism is independent of the topology.

?? Avoid routing loops. Any paths installed by the path update mechanism must not introduce routing loops. Certain protocols may introduce transient loops whilst the update is being processed. This is not necessarily a problem, but the network must be able to converge in a short[40] period of time to a stable routing configuration.

?? Support multiple paths. As discussed earlier in the annex, there are several reasons why being able to make use of multiple paths through the BAN is desirable. Most obviously, it makes packet forwarding more robust. It also allows more efficient use to be made of the network resources in the BAN.

?? Require limited signalling overhead. This is a relatively obvious attribute, but is more of a discriminator between otherwise similar solutions. Any protocol that is a sensible candidate for mobility routing must have a manageable and scalable signalling load.

?? Be robust in the presence of link failures. Link failures need to be detected, either by link-layer messages or an appropriate network layer mechanism and the problem routed around. From a path-update perspective it is assumed that this relates to the failure of fixed network infrastructure. (If the wireless link failed, it is assumed that this would trigger a handover).

?? Be robust in the presence of node failures. Essentially the same problem as link failures. Not only should the protocol be able to detect and avoid failures, but it should also be possible to detect recovery (or new nodes/links) and take advantage of them.

?? Allow separation of local from global mobility. This is a transparency issue – it is an important architectural principle that the BAN is transparent. Thus any global mobility mechanism must not interact with the BAN internal mobility management. It is not directly related to path updates, but it is important that path updates should not compromise this principle. (For example, HMIP would provide local mobility within the BAN, but with the assumption that Mobile IP was being used for global mobility).

?? Install efficient paths. This might initially appear to be a slightly weak statement. However, work on protocols such as MER-TORA [A3.36] has indicated that optimal routing may be expensive (both computationally and in terms of signalling). Relaxing the need for optimality allows efficient routes to be found at a much lower cost. Within the BAN, it is generally desirable to think more abstractly in terms of 'least cost' routes, rather than 'shortest path'. Even so, it is not anticipated that paths should necessarily be optimal – only 'efficient'.

[40] In this sense 'short' is relative to the handover rate, otherwise the network routing will be unable to converge

A3.3.5.2 Possible Solutions

Where the topology of entities which actively participate in path updates is non-trivial, then it would be hard to make a definite statement about a suitable strategy. So, if the BAN were intended to make use of a partially meshed collection of BRAIN-aware routers, it is not clear what scheme should be used. Cellular IP and HAWAII are the classic examples of per-host forwarding protocols; more recent work has considered protocols based on *ad-hoc* networking, notably AODV (Ad-hoc On-demand Distance Vector) and MER-TORA [A3.36]. These can only be considered on a case-by-case basis, assessing the both the protocol and the proposed topology against the criteria outlined above.

We reject any topology based on a strict tree (whether built on tunnels or per-host forwarding) as essentially unsuitable, as this lacks any inherent robustness.

Ultimately this has lead BRAIN to consider a simple, gateway-centric approach as the most easily definable. The details are described in the BRAIN Candidate Mobility Protocol annex (section A3.4.8)

A3.3.6 Path Updates References

[A3.31] Eardley P, Mihailovic M, Suihko T "A Framework for the Evaluation of IP Mobility Protocols", 11th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications. PIMRC 2000, 18-21 September 2000, IEEE, pp 451-7

[A3.32] C. Keszei, J. Manner, Z. Turanyi, A. Valko, "Mobility Management and QoS in BRAIN Access Networks" Ist Interantionsl BRAIN Workshop, London, Novemebr 2000.

[A3.33] Scalability and Resilience, M. West, R. Hancock (SM031)

[A3.34] A Survey on IETF Handover Proposals, T. Suihko (NOK221

[A3.35] S. Kim, Z. Turanyi, A. Valko, "Cellular IP" draft-ietf-mobileip-cellularip-00.txt

[A3.36] S. Corson, "Edge Mobility Architecture" draft-oneill-ema-01.txt

[A3.37] C. Castelluccia, "A Hierarchical Mobile IPv6 Proposal" INRIA Report No. 0226

[A3.38] R. Ramjee, *et al*, "IP micro-mobility support using HAWAII"

A3.4 Scalability and Resilience

Key issues for a BRAIN Access Network (BAN) are those of scalability and resilience. BRAIN is considering access networks over a wide range of sizes from small, localised BANs, through campus-network deployments to large scale networks. Given this, it is important that the implications of providing a scalable, resilient solution are understood.

In general, this annex considers the distribution of functionality across multiple network elements as a way of achieving this goal.

An example topology is shown (Figure A3-24), which gives the top-level relation of the BAN to other parts of the overall network.

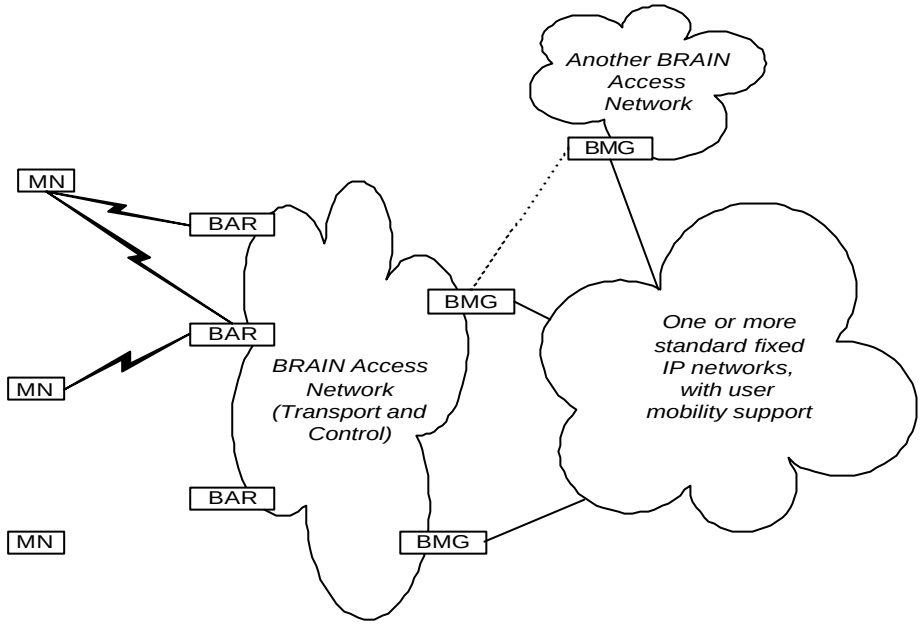


Figure A3-24: BRAIN Network Topology

The key elements of the architecture are listed below with their basic functionalities:

- **Mobile Node (MN)**: IP host with one or more IP addresses and a single interface and possibly more than one simultaneous radio link with different BARs
- **BRAIN Access Router (BAR)**: an IP router with multiple wireless and wired interfaces
- **BRAIN Access Network (BAN)**: data transmission infrastructure and control entities for routing and determining user access
- **BRAIN Mobility Gateway (BMG)**: special purpose IP router hiding any BRAIN-specific routing functionality

A3.4.1 Failure of Nodes

This section discusses the implications of failure of various types of nodes. This identifies where particular support for resilience needs to be engineered into the BRAIN architecture.

In most cases where resilience is required, it may be possible to offer a suitable solution by engineering an appropriately high-availability element. Given related arguments about scalability, it is considered that achieving resilience through distributed functionality is highly desirable. However, it should be made clear that in some cases a highly-available system may be a viable alternative.

A3.4.1.1 Mobile Node

Failure of the MN will clearly prevent communications to and from it. However, there should be no further impact on the rest of the network. If the network makes any use of hard-state information, then this disconnection of the MN needs to be detected by the network. Only then can the appropriate actions be taken to purge this now obsolete information. Alternatively, information may be soft-state, in which case it is assumed that it will time-out after a suitable period of time. BRAIN Access Router

The BAR is the network layer device that mediates between the fixed part of the BAN and the MNs that are connected to it. Failure of a BAR will effectively disconnect all MNs that are currently attached to it. This can be overcome by having any MN in this situation perform an unplanned handover to reattach to an alternative BAR. There is a network planning and deployment issue here to ensure that there is always a suitable alternative BAR that can be used. This also suggests a requirement on the handover function in that it should be possible to complete an unplanned handover without having any communication from the failed BAR (with the attendant security implications). Given that failures of this sort are not expected to be common, an alternative is to allow the mobile to re-login through a new BAR; some 'logout' functionality may need to be triggered by this.

On the BAN internal side, failure of a BAR should be handled as for any BAN internal routing node. Since it is expected that any MN attached to the failed BAR will find an alternative point of attachment, this will cause any per-host state in the network to be updated and consequently implicitly prevent the BAR from being used.

It may be that some network function requires a BAR to hold information about MNs that are not directly connected to it (for example, paging information for an idle MN). This has two clear implications: firstly that there should be some alternative storage in the network; and secondly that there must be a way for the failure of the BAR to be discovered. (This last is a more subtle point than the failure being detected by neighbouring routers, as a distant node may trigger the use of the stored information).

A3.4.1.2 BRAIN Mobility Gateway

BMG failure would be serious, unless a suitable failover strategy was available. Many proposed micro-mobility solutions are unclear about ways in which multiple gateways could be supported, and so this has been a focus of the mobility research within BRAIN. This is discussed fully in section A3.4.3.

(For the purposes of this discussion, the Anchor Point (ANP) of the BRAIN Candidate Mobility Protocol is treated in the same way as a BMG).

A3.4.1.3 BAN Internal Node

A BAN internal node is responsible for maintaining routing information both for fixed network and MN reachability. (This may be maintained by one single or two independent protocols). However this is managed, it is important that the failure of a single node does not affect the ability of the BAN to forward packets to and from a MN.

For a per-host forwarding (PHF), or hop-by-hop type solution, mobile routing information is typically distributed across the whole network. The routing protocol that is used to install this information may handle this reactively or proactively.

In the reactive case, the failure of a node needs to be detected, initially by its neighbours. This information will be propagated by whatever method the protocol defines in order to build a path around the failed node. For the proactive case, multiple routes are found and installed in advance. The failure still needs to be detected but, when this happens, the alternative path already exists; the network simply needs to use it.

In a gateway-centric solution, it is assumed that tunnels are built between nodes in the BAN. Only these nodes are 'BRAIN-aware', and only these nodes maintain per-host state information. All other nodes can be traditional IP routers. A simplifying assumption is that, in this case, the only BRAIN specific nodes are the BMGs and the BARs [41]. Then, any other internal node is a standard IP router and its failure will be handled by whatever routing protocol is in operation.

A3.4.2 Scalability

The concept of scalability can be considered to mean different things in different contexts. The three issues that are discussed in this section are:

?? MN scalability
The ability of the BAN to manage a large number of mobile nodes

[41] The BRAIN Candidate Mobility Protocol introduces the Anchor Point (ANP) as a BRAIN specific entity. It essentially fulfils the definition of BMG as stated later, but without the implication that it is an 'edge' device. The discussion of multiple BMGs discusses this in more detail.

?? Throughput scalability
Avoiding bottlenecks in the network to allow a high volume of traffic flow

?? Geographic scalability
Scaling a BAN to provide coverage over a wide area

Many of these issues are related, but each may have slightly different implications.

A3.4.2.1 Terminal Scalability

Ultimately, scalability to a large number of mobile nodes is limited by the amount of per-host state information in the network. In a per-host-forwarding scheme this can be significantly reduced in general by making use of address aggregation. This means that only when a mobile is attached to a point other than its 'home' BAR is there any need to inject host specific state. For a gateway-centric approach, there will be per-host state for every attached MN at some point in the network. Whatever these elements are (they are assumed to be BMGs in the BRAIN architecture), multiple instances are required to provide scalability. As pointed out at the start of the section, scalability issues inter-relate, and scaling to a large number of terminals would cause an increase in the amount of mobility signalling in the network. This could mean that throughput scalability (in the next section) should take account of the increased load on the network due to signalling.

A3.4.2.2 Throughput Scalability

Throughput scalability can be achieved by deploying additional elements where there are bottlenecks. These may not just be for data traffic flows, but may include signalling. This needs to be taken into account in planning the network and deciding where to add additional capacity. Adding additional fixed network capacity should be relatively straightforward – this requires suitable support for the fixed-network routing. Additional BARs should be deployable with little problem as well, since multiple instances are invariably used. Scalability at the BMG is possible by either deploying multiple instances or building a highly scalable BMG. Given the additional architectural complexity of building a scalable BMG and the other reasons for deploying multiple BMGs, then multiple BMGs are considered to be the preferred solution. (The highly scalable approach is always available should it be appropriate in specific circumstances).

A3.4.2.3 Geographic Scalability

Geographic scalability, to extend the coverage of the BAN to a large physical area, requires more BARs and probably more infrastructure. Since any BAN can manage multiple BARs, scaling a network to add more should not pose any problems. Likewise, scalability of infrastructure should not pose a problem, providing that the required services (routing, paging, etc.) all scale. With regard to the BMGs, multiple instances are desirable in this case to improve routing efficiency in the BAN. Consider a country-wide BAN, and assume that there is only a single BMG. Since all traffic into (and out of) the BAN passes through this node this architecture can cause serious routing problems. The only practical solution to this problem is to deploy multiple BMGs.

A3.4.3 Multiple BRAIN Mobility Gateways

A3.4.3.1 Introduction

Effective support for multiple BMGs has been identified as an important aspect for the deployment of BRAIN Access Networks. This section considers some of the specific issues raised by this support.

There are a number of reasons why it is desirable to have a number of mobility gateways. Perhaps the most obvious is the avoidance of a single point of failure, although resilience is probably the hardest problem to solve. However it is also useful to be able to support a number of mobility gateways for scalability. This applies both to scalability to number of users (by spreading the load across a number of gateways) and to geographic scalability. In the latter case it is more a means of providing optimal (or better) routing through multiple peering points.

With the exception of small BANs, support for multiple BMGs is important. Once this has been accepted, then there are a number of issues raised by having multiple gateways that need to be addressed.

This section addresses the multiple-BMG issue based on the architecture diagram (Figure A3-24). A later section addresses the implications of the Anchor Point introduced in the BRAIN Candidate Mobility Protocol.

A3.4.3.2 Basic Issues

The following sections address more specific issues regarding support for multiple BMGs. For the purposes of this discussion the following simple architecture (Figure A3-25) is assumed.

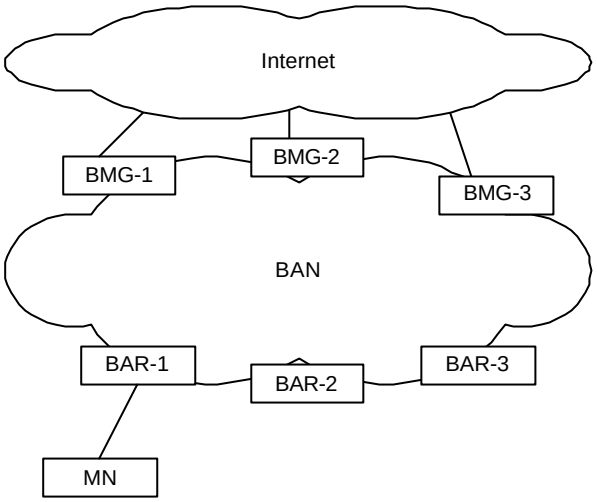


Figure A3-25: Example multi-BMG BAN

A3.4.3.3 Addressing

There are at least two distinct models of address assignment to the MN in this example. It is useful to summarise two obvious alternatives. In the edge-mobility model [A3.39], it is likely that addresses are allocated by entities co-located with or close to the BAR. This is because the use of aggregation for routing efficiency is applied across the whole network and the 'home BAR' is identified by the address block that it owns.

In contrast, with a gateway-centric view, addresses are owned by the gateways. (For this discussion, we assume the BMG). Thus gateways (there typically being fewer of these than there are BARs) manage larger blocks of addresses. A gateway stores the mapping between the address of the MN and the BAR (or some 'next hop gateway), so 'ownership' of BARs is more of an administrative issue.

A3.4.3.4 Gateway to BAR mapping

It is useful to introduce the concept of a BMG 'owning' a BAR – packets may only flow between a BMG and a BAR where the BAR is owned by a BMG. This is a purely administrative concept that described how the network is organised. It also allows for the implications of this mapping to be considered.

The mapping between BMG's and BAR's can be 1:many or many:many. At one extreme a BAR may be owned by exactly one BMG. In this case, handing-off between BARs may force a change of BMG. However it means that the ingress/egress point is always known, or at least can be derived from the BAR to which the mobile is attached. If this constraint is relaxed, then BARs may be 'owned' by many BMGs. Ultimately this can be extended such that every BAR is associated with every BMG. The implications of these approaches are discussed later.

A3.4.3.5 Location updates

When a location or routing update occurs for a mobile node, it may be that one or more BMG needs to be updated. In the event that an update has propagated to a BMG, it may not necessarily terminate at this point. (The update cannot be seen 'outside' of the BMG – it exists to act as a termination point for BAN internal mobility signalling). However, it may be that the information should be shared with other BMGs.

Some solutions, most notably those built on per-host forwarding protocols may automatically provide this functionality. It is more likely to be true in this case as routing information is distributed across a subset of the nodes as defined by the protocol. If the protocol supports multiple gateway nodes, then updates should take this into account. In the event that the protocol does not support multiple gateways, it may be difficult to extend it – the necessary updates would fall outside of the normal scope of the protocol.

While there will frequently be situations in which the BMG does not need to be informed at all, this is not necessarily always true. Whenever the mobile crosses into a 'shared' region (i.e. is attached to a BAR that is owned by more than one BMG) then it may become desirable to pass location/routing information to the BMG.

A3.4.3.6 Handover

The most obvious form of handover within the BAN is where the MN changes BAR. However, if there are multiple BMGs then it is possible handovers could also occur between BMGs. This may be correlated with inter-BAR handover, or a separate event.

This is discussed more fully in section A3.4.5

A3.4.3.7 Resilience

Resilience is one of the key reasons why multiple BMGs need to be supported. However, it is also one of the more complex issues. Resilience can only be achieved if each BAR is owned by more than one BMG, in order that a switch can be made in the event of failure.

A3.4.4 Data flows

A3.4.4.1 Downlink

For any given downlink flow, it may be desirable for the ingress BMG to be known to the network. This is particularly true, for example, where RSVP proxies are being used. In order that the correct end-point is chosen the ingress BMG for the flow should be known. The situation may be relatively clear for a specific connection, but may be subsequently complicated. For example, new flows may enter the BAN at the same or possibly a different BMG. Yet more complex is the possibility that the ingress BMG may change during a flow (this would need to be detected and signalled appropriately).

This example is a specific instance of a general question regarding the routes that are advertised by the BMG. Specifically, the visibility of addresses through route advertisements is critical in establishing the downlink behaviour. The question can be formulated as a number of possible options for the advertisement/ownership of IP addresses used by MNs. They may be:

?? Owned solely by a specific BMG
   This does not allow for robustness, other than by reassignment of the MN IP address (which does potentially offer robustness, but not seamlessly so). It does effectively facilitate scalability.

?? Owned equally by all BMGs
   This is clearly very robust, but does not easily facilitate scalability. (This is true in the general case – if host-specific routes are rarely propagated to this level in the network, then it is less of an issue).

?? Preferentially owned by one BMG, but known to all (or many)
   This is closer to the classical fixed IP method of advertising routes. It allows the BAN to control the ingress point for a given destination, but also provides a high degree of robustness.

Any of these models are deployable candidates. However, it is the last one that offers the most flexibility.

A3.4.4.2 Uplink

The fundamental issue affecting uplink is whether the packets can exit the network at any BMG, or whether they are constrained. Whether or how the egress can or may change is essentially the same as for the downlink case. The network needs to decide on which egress point to use. This may be considered as an issue of flow-symmetry, or as how routes learned from outside the BAN are handled.

A3.4.4.3 Flow symmetry

In general, when there is a choice of BMG for ingress and egress, then the possibility of asymmetric routes through the network exists. If this were considered problematic, then the intra-BAN mobility solution would need to address this. With a tunnel-based solution, reverse tunnelling provides an obvious

way of managing this. A per-host-forwarding system is less likely to offer this (since uplink flows and downlink flows are routed to different destinations and the mobile may be distant from its home location).

A3.4.5 Inter-BMG handover

As the mobile node wanders and changes point of attachment to the BAN the optimal ingress point may change. The sample architecture outlined above (Figure A3-25) and used here can be assumed to have the following basic mapping between BAR and BMG. That is, each BMG owns the BAR with the same number. For a consideration of handover, details of redundancy are not important.

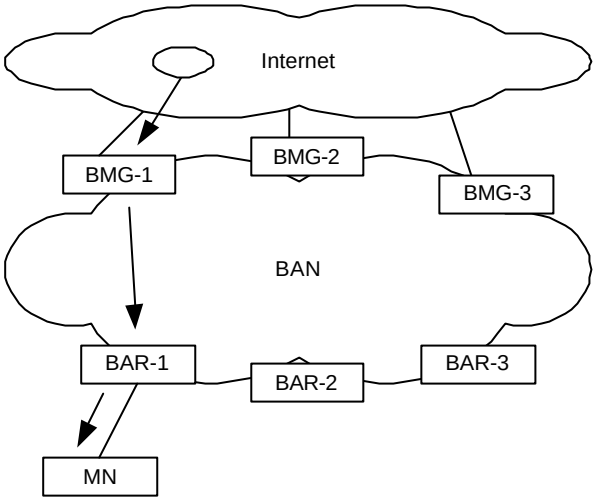


Figure A3-26: BAN with Attached MN

For example, in the diagram above (Figure A3-26), the routing is through BMG-1 to BAR-1. If the mobile changes point-of-attachment, perhaps arriving at BAR-3, the route across the BAN is sub-optimal. (How this routing might be managed is discussed later). Note that this is primarily a concern of the routing within the BAN – it may be that the path across the Internet is sufficiently long that the portion across the BAN makes little difference end-to-end. It also seems reasonable that as a criterion for establishing the 'best' route, that 'least cost' is the goal (rather necessarily than smallest number of hops).

If, by whatever criteria, routing from the BMG to the BAR needs to be optimised, then the mobility management solution needs to consider how to update the appropriate BMGs with sufficient information to make this possible. This also needs to happen in a way that avoids routing loops.

The ingress point can only be affected by the routes that the BAN advertises. Although theoretically this means that the routing for a single MN can be fixed by changing the advertisement, this is utterly impractical. Since it would require leaking per-host information into the Internet it is both architecturally bad and completely unscalable.

The implication of this is that the only way to manage this situation is to allow (from the BAN's perspective, encourage) the MN to change IP address. This will automatically cause packets addressed to the new IP address to be passed through the most appropriate BMG.

The most logical way to consider a change of address in this form is as a handover between BMGs. Where handover is required between BMGs, then the characteristics of such a handover should be considered. Particularly of interest is how smooth the handover is, or needs to be. It should be recognised that if the boundary between BMGs is hard, i.e. handover between a particular pair of BARs forces BMG handover, then it is important that the handover is smooth (This is a consequence of a 1:many mapping between BMG and BARs). If, however, there is overlap and the boundary between BMGs is blurred, then it may be less important that BMG handover is smooth. Since the BMG handovers can be de-correlated with inter-BAR handover, then the MN has the opportunity to select a better (if not the best) time to carry it out. Ideally, this would only happen when the MN had no active sessions, so issues of smoothness are unimportant.

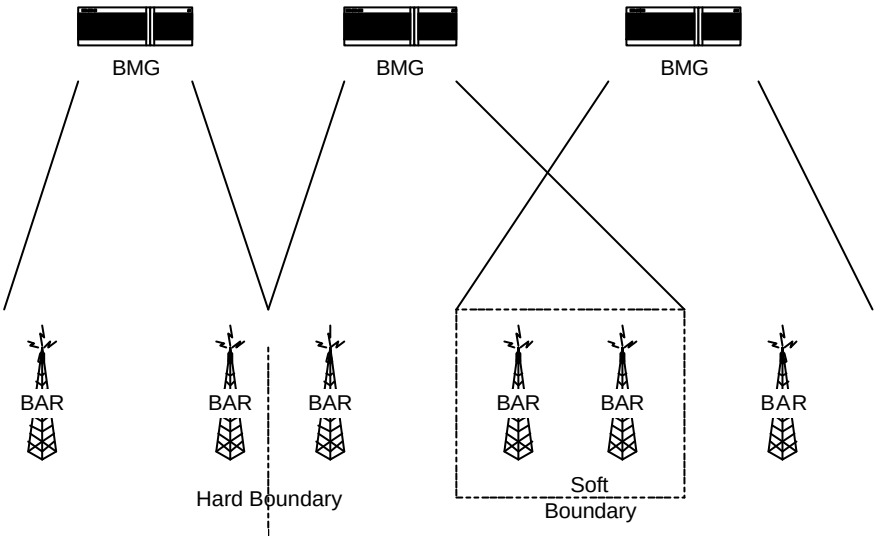


Figure A3-27: BMG/BAR Boundaries

A3.4.6 Multi-BMG Routing

This section presents some diagrams to clarify options for routing within the BAN where multiple BMGs are present. It is not intended to be exhaustive, merely to illustrate some possible scenarios.

The simple architecture diagram (Figure A3-26) gives the basic concept of a BAN with 3 BMGs and a number of BARs, 3 of which are shown. (It is assumed that there are many more BARs, but that this is sufficient to consider the scenarios).

Initially, where the MN has just attached to BAR-1, it is assumed that BMG-1 will be the natural ingress and egress point. This means that routing initially follows this path:

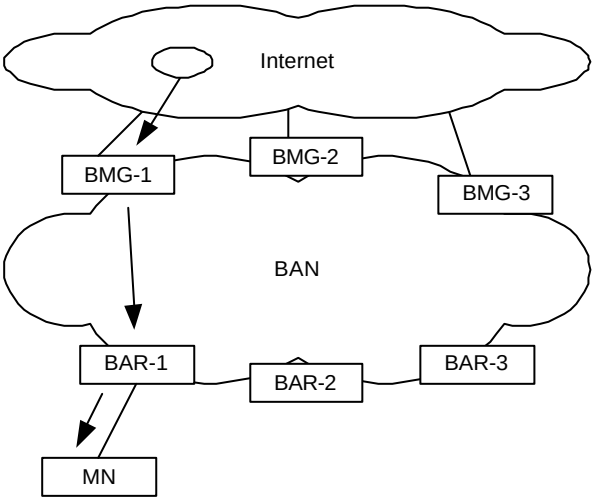


Figure A3-28: Example Traffic Flow

Note that this is based on the assumption that basis for choosing the BMG is within the BAN (i.e. the location or, more likely, the address of the MN) rather than of the correspondent node. Otherwise, assuming that the routing information were available, a correspondent node close to BMG-3 might be routed through BMG-3 (to get packets into the BAN as soon as possible). As the MN moves through

BAR-2 on towards BAR-3, the route is less obvious. The following diagrams suggest some options for how this could be managed.

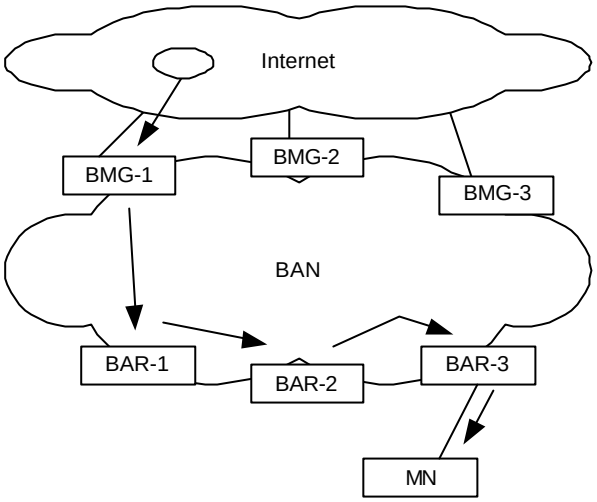


Figure A3-29: Host Moves to BAR-3; case 1

This first case assumes that the BMG remains the same, so implying that the address of the MN (even if also 'owned' by another BMG) is also advertised through BMG-1. This is the edge-mobility case, where the packet is forwarded around the (far) edge of the access network. Some route optimisations might occur and the MN may (eventually) change its care-of address, which will presumably change its associated BMG.

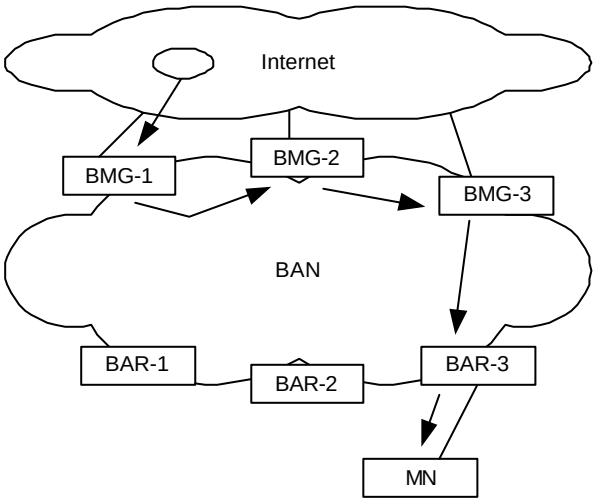


Figure A3-30: Host Moves to BAR-3; case 2

This second case could also be considered edge mobility, but here the routing is along the ingress edge, to reach the 'best' BMG for the current point of attachment of the mobile node. This clearly means that routing / location information from the mobile node need to be propagated to (at least a subset of) the BMGs.

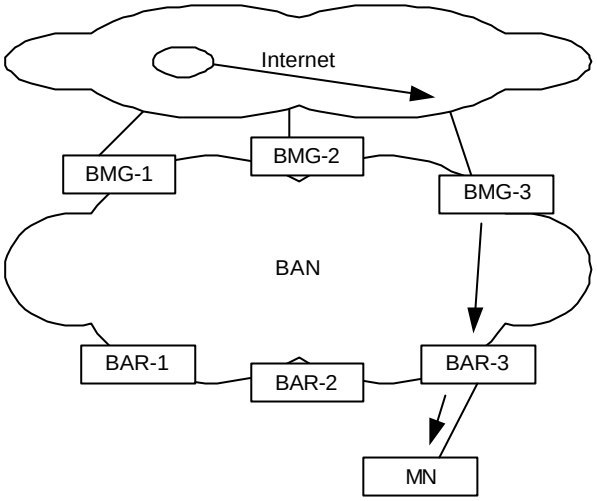


Figure A3-31: Host Moves to BAR-3; case 3

Finally, this diagram considers the case where the ingress BMG changes to optimise (at least from the point of view of the BAN) the route. This can only be achieved by changing the CoA of the MN, or by otherwise changing advertised routes. The latter option is unlikely to be feasible, as it is advertising host routes outside the BAN. (This is clearly possible, but unwise). Essentially, this describes the routing that would be expected after an inter-BMG handover.

A3.4.7 BMG Routing

As regards routing by the BMG, this may be considered as a special case (potentially with its own protocol) or as part of the general mobility solution. The characteristics of routing at the BMG level may be considered quite different from general routing within the BAN.

If BMGs are meshed together at the 'top' level of the BAN then routing within this mesh might still be treated in a number of different ways:

?? Optimal routing requires that the route information is updated whenever a MN crosses a boundary between BMGs

?? Static routing would allow for some resilience

?? A solution could be intermediate; somewhere between these 2 extremes.

Possible options for routing between BMGs are:

?? Simply handled as part of the general mobility protocol

?? MER-TORA offering some form of edge mobility

?? 'Truncated' Cellular-IP or HAWAII

(using a tree, but considering the nodes that would logically be 'above' the BMGs to be virtual nodes[42])

?? An alternative mobility protocol

A3.4.8 BRAIN Candidate Mobility Protocol

A significant part of this document discusses some of the issues surrounding the support of multiple BRAIN Mobility Gateways. The BRAIN Candidate Mobility Protocol [A3.40] introduces the concept of an Anchor Point (ANP) and treats the BMG as an essentially standard border router.

[42] Rather than specification of a routing protocol, this is more of an abstraction for organising the BMGs. Thus it is more a way of logically thinking about how packets may be exchanged.

The scalability and resilience requirements of the BMG in this model are clearly addressed by the use of standard fixed network routing protocols. The BMG also has no mobility specific functionality. The ANP then becomes the termination point of the mobility related signalling in the network. That is, it is the ANP that makes the BAN look like a normal, wired, IP network.

Thus, to provide continuity between the discussion in this document and the architecture in [A3.40], it should be assumed that the BCMP ANP is equivalent to the original definition of BMG provided at the start of the document.

The decoupling of the boundary and mobility specific functionality allows for more flexibility in the selection and deployment of network components. It also makes an explicit boundary that looks like a fixed-network boundary. Fundamentally, it does not change the original architecture, other than the minor re-naming.

A3.4.9 Scalability and Resilience References

[A3.39] S. Corson, "Edge Mobility Architecture" draft-oneill-ema-01.txt

[A3.40] C. Keszei, et al, "Description of the BRAIN 'Candidate' MM Protocol", WP2-ER014-0_3-pi

A3.5 BRAIN Candidate Mobility Protocol Components

The messages of this protocol can be categorised into two groups. The first group includes all the messaging between the MNs and the BARs representing the edge of the BAN. These messages are always exchanged using the air interface so with certain radio link layers it might be possible to use existing L2 primitives instead of IP packets to improve performance. Such adoptions of this protocol to certain link layers are outside of the scope of this document and should be separately defined. It is important to note that such use of L2 primitives are allowed only if it does not affect the semantics and operation of the protocol.

The second group of messages consists of the messages inside the BAN. Here all messages are IP messages. The source and destination IP addresses are set according to the senders and receivers of the message. For example when a BAR forwards a HOFF message to an Anchor, then the source address is filled according to the BAR's (uplink) IP address and the destination address is set to the Anchor's IP address. The mandatory *session id* field in the messages identifies the MN that the message refers to. The fact that signalling messages between one particular BAR and Anchor has fixed source/destination addresses (i.e., not the MN's address used to identify the originator) helps e.g., the identification and QoS categorisation of the signalling messages.

In the following sections we give the brief description of the used procedures and protocol phases.

A3.5.1 Initial Login

A newly arrived MN must log into the BAN. The purpose of the login procedure is to authenticate and authorise the mobile host, and obtain an IP address. To log in the mobile node sends an LREQ message to the BAR it has radio connection to. The *global id* field of this message is used to identify the MN for AAA purposes. For example, it can be a NAI string, like user@domain. The next field, *security info* is provided by the MN and will be used to authenticate it during the global AAA process. The whole message is formed similar to a DHCP request that is, the source IP address field is set to zero.

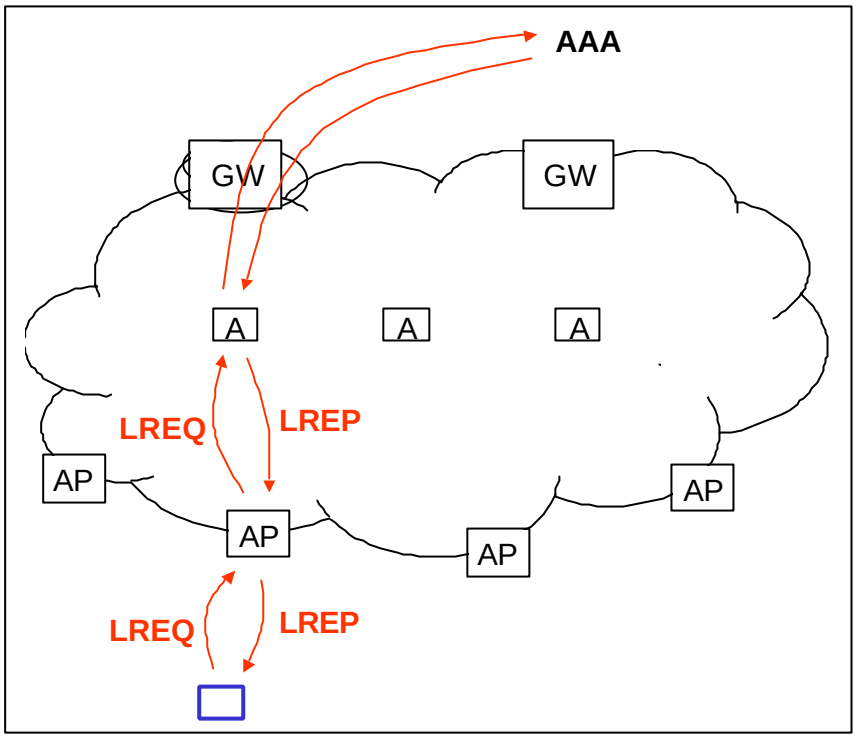
LREQ		
fields:	present	remarks
global id	A	
security info	A	

The BAR stores the MAC identity and the *global id* of the MN and forwards the LREQ message to a selected Anchor. The Anchor is selected according to operator specified policies. This function of the BAR is similar to that of a DHCP proxy agent. When the Anchor receives the LREQ message it uses an external AAA protocol or local database to get information about the MN and verify its identity.

In response to the LREQ the Anchor generates an LREP message that is sent back to the BAR. If the authentication was successful, the message contains an *IP address, session id* and the encrypted *session key* for the mobile host and its initial *paging area*. The *session id* is used in all further messages to identify the MN. This is a unique but not permanent identifier, which also identifies the issuing Anchor. The *session key* is used to authenticate the MN's further messages. If the authentication failed the LREP message contains the reason in the *result* field.

LREP		
fields:	present	remarks
result	A	
global id	A	
session id	O	
session key	O	
IP address	O	
paging area	O	see Section A3.5.6

When the BAR receives the LREP message it looks up the MAC identity of the MN based on its *global id* and forwards the message to it. By receiving an LREP message from the BAR indicating a successful authentication, the MN is logged in.



A3.5.2 Handover Preparation

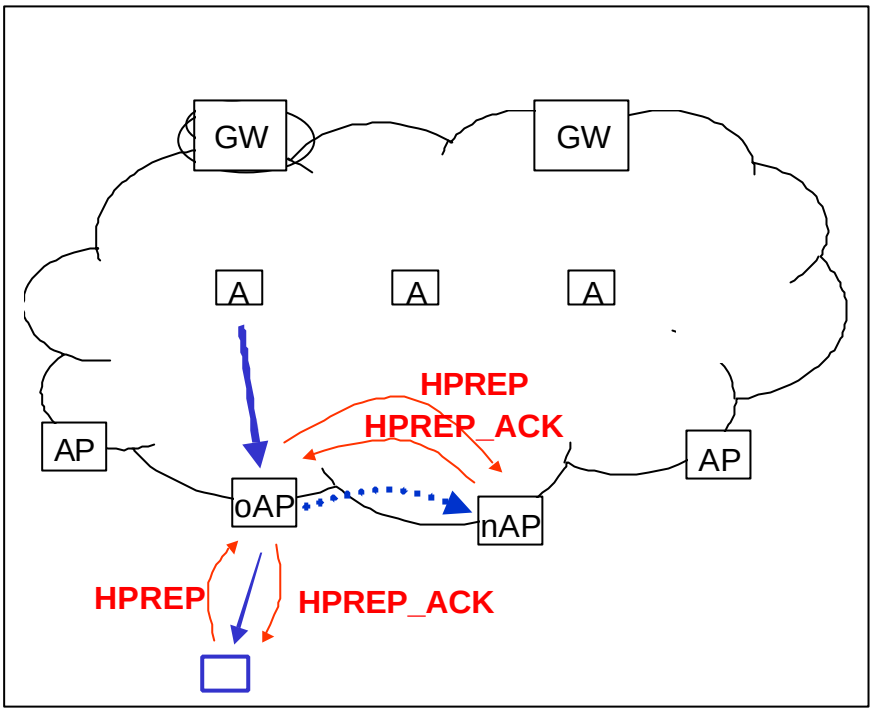
This function is used to inform the BAN about a planned handover. This process is not mandatory for all handovers; it is intended to make handovers seamless.

The MN sends a HPREP message to its old BAR indicating the purposed new BAR in the *nBAR* field. The old BAR forwards this message to the candidate BAR which replies with a HPREP_ACK message. The new BAR indicates its willingness to accept the MN in the *result* field. If the answer is positive the old BAR builds a temporary tunnel towards the new one on the reception of the HPREP_ACK message. Any packet arriving to the MN is also sent to the new BAR through this tunnel. When the old BAR receives the HPREP_ACK message it forwards it to the MN. If network policy permits the MN can be allowed to set up tunnels to more than one BAR at the same time. These tunnels are removed after a timeout or after handover execution. If the MN has been rejected by the new BAR it may try to prepare a handover to another one.

HPREP		
fields:	present	remarks
session id	A	may be implicit between MN and AP on L2
nBAR	A	may be implicit between ARs
L2 parameters	O	
QoS context	O	

HPREP_ACK		
fields:	present	remarks
session id	A	may be implicit between MN and AP on L2
result	A	

The *L2 parameters* and *QoS context* fields of the HPREP message are not generated by the MN but rather appended by the old BAR. These fields are used for context transfer from the old BAR to the new one.



A3.5.3 Handover Execution

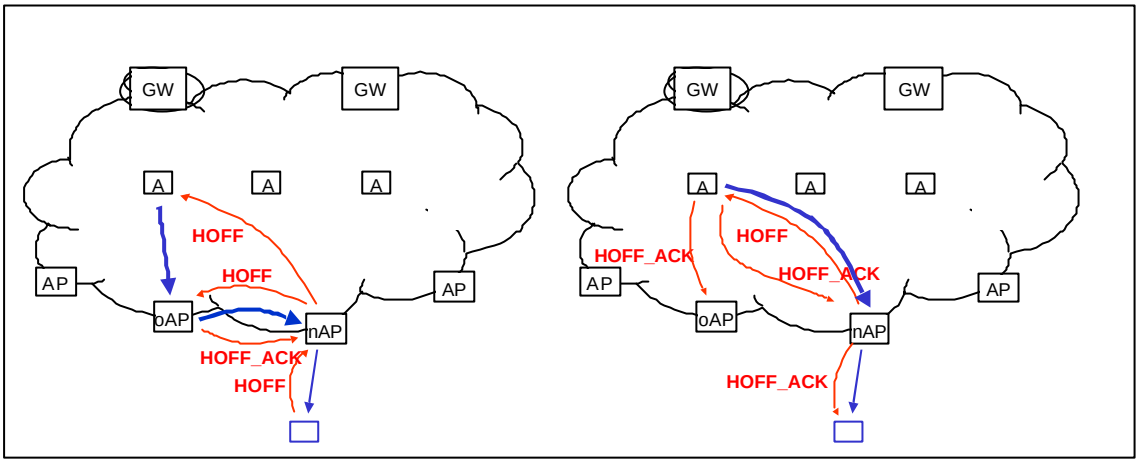
This function is used to perform a BAR change (handover) and in practice it means location update rather than a radio function. The MN sends a HOFF message to the new BAR after a successful L2 handover. The *oBAR* field of this message shows the MN's old BAR. After receiving the HOFF message the new BAR contacts the old BAR by forwarding the HOFF message to it and retrieves all necessary information about the MN in the HOFF_ACK message. In case of unplanned handovers, this is the first time the two BARs get in contact and they can start building a temporary tunnel similar to the planned case (see Section A3.5.2). The HOFF_ACK message between the old and new BARs also used for QoS context transfer in case of unplanned handovers.

In case of planned handover the tunnel is already in place. The HOFF message between the two BARs is only used to release radio resources at the old AP.

In addition to forwarding the HOFF to the old BAR, the new BAR forwards the HOFF message to MN's Anchor Point. The identity of the Anchor is determined from the *session id* (see Section A3.5.1). The field *Anchor info* contains information about possibly better Anchors for the MN and is used only if the BAR has suggestions for the Anchor. In response the Anchor configures a new tunnel toward the new BAR and replies with HOFF_ACK message. From this point on all traffic addressed to the MN will be forwarded in the new tunnel towards the new BAR. The new BAR forwards the HOFF_ACK message to the MN. The *IP addr (w)* field (filled in by the Anchor) may contain a Anchor change suggestion with a weight value. The weight indicates Anchor change urgency, i.e., higher values mean higher urgency. The old BAR also receives a HOFF_ACK message from the Anchor in order to release all resources allocated to the MN, including the temporary tunnel.

HOFF		
fields:	present	remarks
session id	A	
Obar	A	may be implicit between BARs
paging area	A	see Section A3.5.6
Anchor info	O	only between new BAR and Anchor

HOFF_ACK		
fields:	present	remarks
session id	A	
Result	A	
IP addr (w)	O	not present between BARs
L2 parameters	O	
QoS context	O	



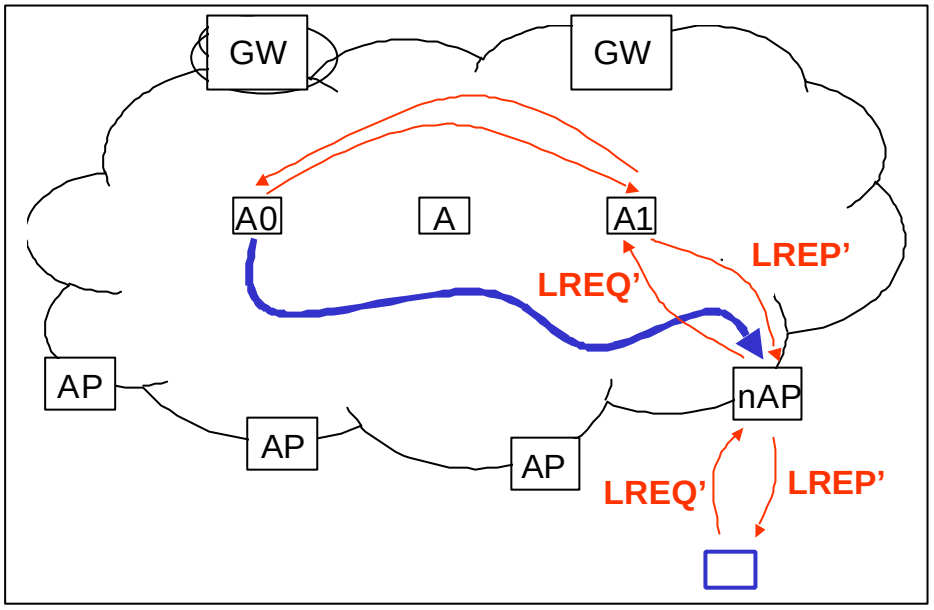
A3.5.4 Change Anchor Point

After a handover the network may suggest the MN to switch to another Anchor to decrease triangular routing. Such a switch results in the change of MN's IP address, which breaks ongoing sessions. Therefore, the MN is left to decide the exact moment of the change and to initiate the procedure. The Anchor change is very similar to the Initial Login (see Section A3.5.1). The most important difference is that the MN already has a session id and IP address and is authenticated locally by the BAN, instead of the AAA infrastructure. The latter is important because global AAA procedures are not needed in this case.

The MN sends an LREQ message to its actual BAR. This message is the same LREQ message as the one mentioned at Section A3.5.1, except that the optional *session id* field is present this time. The BAR forwards it to the best Anchor in the current situation. The candidate Anchor then exchanges all necessary information with the old Anchor, which is determined from the *session id*. As a result, the old IP address and session id is freed. These resources, especially the session id, should not be immediately allocated for another MN. The new Anchor allocates the *new session id, new session key, new IP address* for the MN and sends it in an LREP message to the actual BAR. The MN is still identified by its old *session id* so the message must contain it too. The BAR updates the MN's record and forwards the LREP message to the MN. If the authentication of the MN failed the new Anchor may reject the re-login request and the LREP message contains the reason in its *result* field.

LREQ		
fields:	present	remarks
session id	O	not present in case of Initial Login
global id	A	
security info	O	mandatory in case of Initial Login
challenge	O	

LREP		
fields:	present	remarks
session id	O	not present in case of Initial Login
result	A	
global id	A	
new session id	O	
new session key	O	
new IP address	O	
paging area	O	see Section A3.5.6

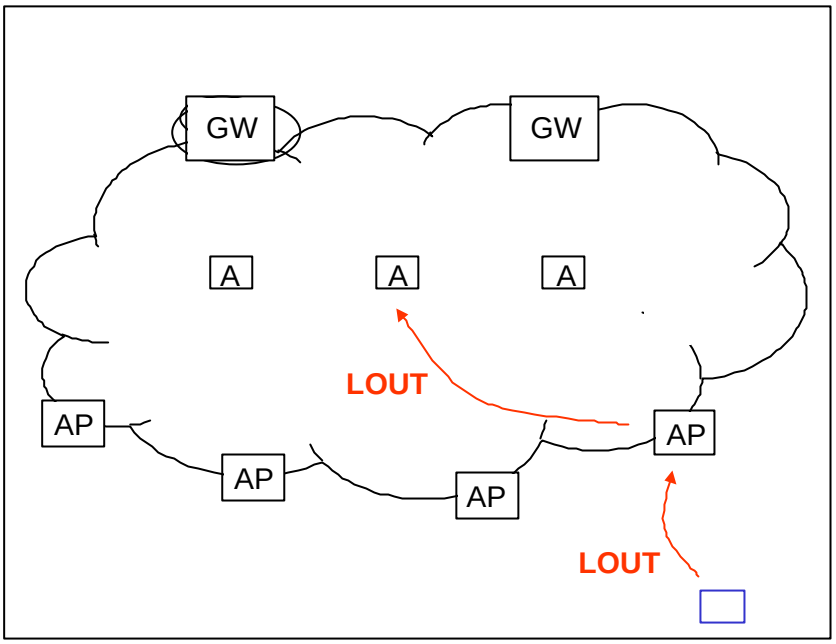


A3.5.5 Logout

To log out from the BRAIN network the MN sends a LOUT message to its actual BAR. The BAR will forward it to the MN's Anchor and it releases all resources. The Anchor will free the MN's *session id* and IP address. This is an unacknowledged message.

This message may be used by the Anchor in order to explicitly remove the MN from the BAN. In this case the Anchor sends the LOUT message to the MN's actual BAR and fills in the *reason* filed of the message. The BAR may inform the MN by forwarding this message.

LOUT		
fields:	present	remarks
session id	A	may be implicit between MN and AP
reason	O	



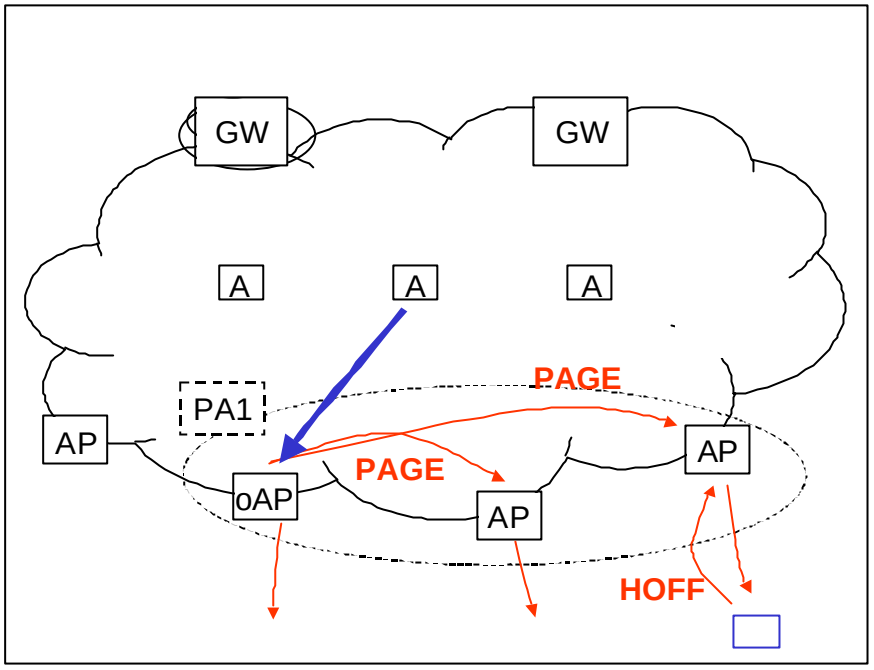
A3.5.6 Paging Mechanism

Idle mode means that the MN is allowed to move in the BAN without location signalling while it is inside a paging area. The paging areas might be overlapping which means one BAR may belong to several paging areas at the same time. During Initial Login the *paging area* field of the LREP message tells the MN its initial paging area. During handovers the MN chooses one among the paging areas the new BAR is belonging to and tells it in the *paging area* field of the HOFF message. This field is present only in those messages, which are sent between the MN and the BAR, because inside the BAN this information is not needed. The BAR at which the MN went idle, is called the last BAR (lBAR). It is the only one device, which knows about the MN's idle state.

If an IP packet arrives to the Anchor for an idle MN, the Anchor forwards it to the BAR where the tunnel ends, that is the lBAR. The lBAR knows about the idle state of the MN and pages it by sending PAGE messages to the other BARs in the paging area of the MN. In addition, the lBAR buffers incoming data packets. When the other BARs receive the PAGE message they broadcast it over their air interface. At the reception of the paging the paged MN responds with a HOFF message to its closest BAR and performs an unplanned handover to it. If the handover does not happen in a specified amount of time, the lBAR might repeat the PAGE message a couple of times. After several unsuccessful trials the lBAR may give up and send a LOUT message to the MN's Anchor indicating the absence of the MN.

When an idle MN has packets to transmit, it resumes active state. This is done by performing an unplanned handover between the last/old BAR and the actual/new one.

PAGE		
fields:	present	remarks
session id	A	


GW
GW
A
A
A
PAGE
PA1
AP
PAGE
AP
oAP
AP
HOFF

A3.6 Mobility Management and QoS in BRAIN Access Networks (London Workshop Paper)

This section presents a paper co-authored by Csaba Keszei, Jukka Manner , Zoltán Turányi and András Valkó and was presented at the London BRAIN Workshop.

A3.6.1 Abstract

The functionality and efficiency of micro mobility schemes in wireless networks has a great impact on the overall performance of the whole network. In small cell wireless LAN environments, the network must support both frequent and numerous handovers at the same time. Together with providing fast and reliable handoffs, the micro mobility scheme must not affect the service perceived by users. This paper builds on and continues the micro mobility protocol evaluations that have been carried out in the BRAIN project. We select a small number of key design choices to classify and group existing protocol proposals. We observe that with appropriate extensions most existing protocols could be adapted for a BRAIN environment and we make suggestions on these extensions. We also show that some of the extended protocols show fundamental operational similarities when used in a BRAIN environment. As a second goal, we present a QoS architecture framework that would be as independent as possible from the micro mobility scheme used, thus leaving most room for different micro mobility decisions. Having the micro mobility and QoS mechanisms separated allows the parallel and independent evolution of the protocols. We conclude that despite the apparent diversity of existing mobility management protocols BRAIN has a limited set of applicable protocols to choose from and we outline the most important properties of alternatives. Some fundamental design decisions can however affect the QoS architecture. These issues are also identified.

A3.6.2 Introduction

The BRAIN project has investigated a number of IP based micro mobility proposals from various authors [A3.58]. These micro mobility proposals define the method of location update and routing in the environment of a BRAIN Access Network (BAN) that connects the wireless access points to the BRAIN Mobility Gateway (BMG). In addition to the base functionality, many of these protocols contain additional features, such as security support or various seamless handover procedures.

Some of these protocol features present in one protocol can be easily ported into another protocol. Seamless handover support, for example, is implemented in Cellular IP using semi-soft handovers. The basic idea of semi-soft handovers can be applied, for example, in a multi-level Hierarchical Mobile IP network. In this case, the protocol messages of Hierarchical Mobile IP may even remain unchanged, as Mobile IP messages already contain the required flags to support simultaneous bindings. Only the behaviour of Regional Foreign Agents needs to be modified to respond to such simultaneous registrations and temporarily perform the multicasting of downlink data packets to both the new and the old points of attachment.

Many visibly large differences among the candidate protocols result from differing message formats. The similarities between Hawaii and Cellular IP, for example, would seem more apparent if the format of Cellular IP route update packets were similar to Hawaii path set-up messages. This would be possible as both messages play similar function in their respective protocol. The format of the protocol messages is somewhat independent of the protocols' detailed operation and is many times strongly influenced by the history of the protocol and the standardisation process.

There are, however, protocol properties that are fundamental to a particular protocol. These properties constitute the "architecture" of the protocol. They determine the basic operation of it and cannot be "removed" from a protocol or "ported" to another one without fundamentally changing it. When designing the BRAIN protocol, such fundamental properties needs to be decided first as they have the largest impact on the resulting protocol. Additional features such as security support, that are somewhat independent of these properties and can be added to more or less any protocol "architecture". Similar, exact message formats can be determined later based on the architecture, compatibility requirements and the required additional features.

By identifying fundamental properties candidate protocols can be grouped into categories. Protocols belonging to the same category share the same basic behaviour. Due to this similarity it is usually possible to add the same features to all the protocols in the category. This leads to protocols that are essentially the same and can be merged into a single protocol per category. Such merged protocols can form the base of a future BRAIN micro mobility protocol.

From the QoS point of view, the problems of mobility and the mobility-related routing schemes are related to providing the requested service even though the MN changes network connection points. Handovers between base stations, change of IP-addresses, and mechanisms for the intra-domain micro mobility support may create situations where the service assured to the MN cannot be provided, and a QoS violation may happen. A QoS violation may result from excess delays during handovers, packet losses, or even total denial of service. In the case where the user did only request some relative priority to the flows, a short QoS violation may fit within acceptable limits. If the flows were allocated explicit resources, the new network access point and route from the domain edge should provide the same resources.

The domain internal micro mobility schemes may use different tunnelling mechanisms, multicast or adaptable routing algorithms. Tunnels in general can affect the forwarding of QoS-sensitive flows since the original IP-packet is encapsulated within another IP-packet. Multicast can have ill effects on the resource availability, for example, because the multicast group can vary very dynamically; the required resources for assured packet forwarding may change rapidly inside the domain, triggering different QoS-related resource controlling and reservations.

IP address management can also have an effect on providing QoS. In some micro mobility schemes the CoA changes at each handover, which affects the routers' ability to identify the same logical flow. Keeping a steady IP-address within the same domain is therefore an important property of a QoS-friendly micro mobility scheme. Finally, the ability of the micro mobility scheme to "follow" the MN's path and preserve an optimal routing can affect the forwarding quality of packets and result in re-reservations of resources.

The proposed QoS architecture framework aims to define a decoupled architecture in view of micro mobility and QoS protocols. We aim in defining a QoS architecture that would be as independent as possible from the micro mobility architecture; it would neither assume anything from the micro mobility framework and nor would it restrict the implementation and applicability of any of the, possibly modified, micro mobility protocols.

The discussion presented in this paper is structured into two main sections. In Section 2 we study micro mobility management protocols and make suggestions on a BRAIN mobility management protocol. This is followed by a discussion of service quality provision techniques in a BRAIN access network in Section 3. Finally, in Section 4 we present some concluding remarks.

A3.6.3 Mobility Management

In this section we will classify existing micro mobility protocols based on their applicability in BRAIN and will make suggestions on adapting them to a BRAIN access network. The classification will include the following protocol proposals:

?? One level (inside BAN) hierarchic Mobile IP. The Regional Registrations (**RR**) draft defines a Gateway Foreign Agent (GFA) which plays the role of Foreign Agent (FA) in the macro mobility protocol. The mobiles are connecting to Leaf Foreign Agents (LFA) so data packets are tunnelled from the GFA to the LFAs and no intermediate mobility agents are involved. Signalling information in case of LFA change (handover) reaches the GFA in any case.

?? Multi-level Hierarchical Mobile IP extension introduced by Malinen, Castelluccia, Soliman and the appendix of the Regional Registration draft. We will collectively refer to these fundamentally similar protocols as **HMIP**. In these protocols multiple levels of hierarchy are described. They are similar to **RR**, with the important exception that additional mobility agents are placed between the GFA and the LFAs. Data packets are delivered using subsequent tunnels between mobility agents of the various hierarchy levels. Signalling information in case of handovers reaches only the cross-over mobility agent (router) to reduce overhead.

?? Cellular IP [A3.57] (**CIP).**

?? Handoff-Aware Wireless Access Internet Infrastructure [A3.62] (**HAWAII**).

?? Dynamic Source Routing (**DSR**).

?? Ad-hoc On demand Distance Vector [A3.41] (**AODV**).

?? Temporary Ordered Routing Algorithm with Edge Mobility support [A3.60], [A3.74] (**MER-TORA**).

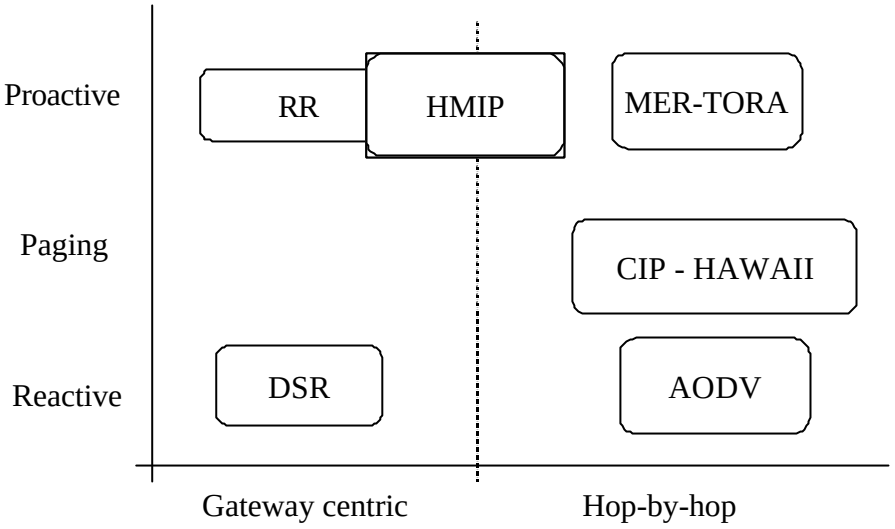


Figure A3-32: Mobility Protocols Classification

A3.6.3.1 Classification of Micro Mobility protocols

Our analysis is based on two fundamental protocol properties related to micro mobility. These two properties are consequences of independent design options. In what follows, we describe the selected properties, followed by a classification of micro mobility protocols in Figure A3-32.

The first property used in our classification is the protocol's *proactive* or *reactive* nature. In a proactive routing protocol the network knows the location of mobile hosts at any time. This is ensured by updating location information upon every movement. In contrast, in a reactive protocol mobile hosts are searched on demand when there is incoming data. In this case before a new session the Mobile Node (MN) must be searched using broadcast messages.

The proactive and reactive approaches represent two extremes of a trade-off inherent in mobility management. Proactive protocols result in very high location update load but they allow immediately routing data to a destination mobile host. Reactive protocols require little or no signalling and processing load when there is no data traffic, but they involve large broadcast search load upon session start-up.

Our next classification criterion is the protocol's gateway centric or hop-by-hop nature. Gateway centric means that the BMG is in the knowledge of the location of all MNs and decides, which BRAIN Access Router (BAR) packets should be routed to. BARs represents attachment points to which mobile hosts can connect to using their wireless interfaces. Since other routers in the network do not have per-host location information, in this case route information must be carried by each packet from the gateway along its path toward the BAR. This implies that gateway centric protocols must use a modified packet format, typically using tunnelling or source route options. In hop-by-hop micro mobility protocols, in contrast, devices along the downlink path make autonomous decisions on the next hop according to the packet's destination address and the per-host databases stored in the network nodes. In this case data can be transported using regular IP packets without optional parameters.

Gateway centric approaches have the advantage that nodes in the access network can be regular routers without specific BRAIN support. These nodes need not have per-host location entries implying that gateway centric protocols can be applied on top of existing networks. On the negative side, gateway centric solutions imply that each location update operation must reach the gateway, which may become a performance bottleneck. This is unlike hop-by-hop protocols that allow location update messages to be discarded before reaching the gateway if the route change does not affect the gateway (see Figure A3-33).

The choice between gateway centric or hop-by-hop nature represents a trade-off between simple, incremental mobility management on the one side and a customised micro mobility network using specialised mobility aware routers on the other side. We believe that the proper choice depends on the

network scenario. In a large network where a single gateway may not cope with the update load the hop-by-hop approach is preferable. If reuse of legacy routers in the network is important then the gateway centric approach should be used.

A3.6.3.2 Protocol Adaptations to BRAIN (Discussion)

While all of the mentioned protocols are applicable to a BRAIN access network, we claim that neither can completely satisfy all of the requirements presented by BRAIN. In what follows, we list a set of protocol features that we believe a BRAIN access network protocol must have. Next, we look at each protocol separately and determine the features that need to be added.

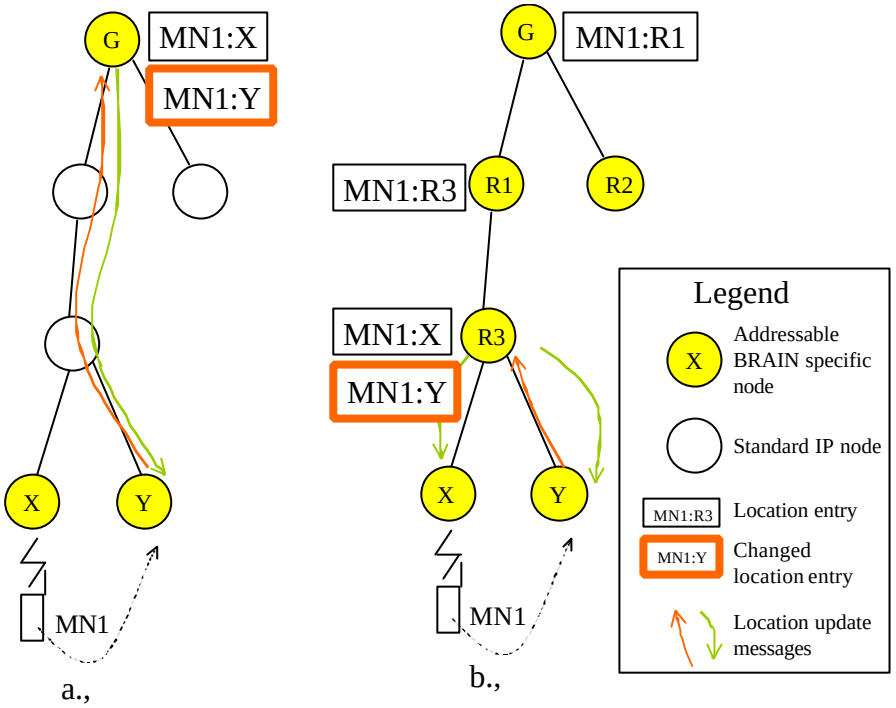


Figure A3-33: Mobility Scenarios

	Paging support	Optimal routing	Address aggregation	Seamless handovers	No single p. of failure	MN-MN	MIP inde p.
RR	-	+	n.a.	+	-	-	-
HMIP	-	+	-	+	-	+	-
CIP	+	+	-	+	-	-	+
HAWAII	+	-	-	+	-	+	-
DSR	-	+	n.a.	-	+	-	+
AODV	-	+	-	-	+	+	+
MER-TORA	-	+	+	+	+	+	+

Table A3-4: Properties of Micro Mobility Protocols

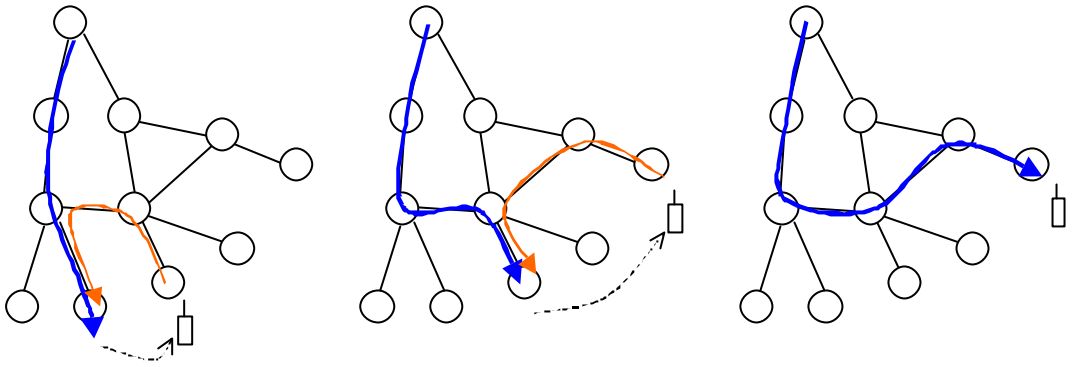


Figure A3-34: Mobile Node Movement

This step will lead to our most important observation. We claim that by extending existing protocols with support for the missing features, the differences found in these protocols are becoming smaller and sometimes negligible. In fact, after making all the necessary modifications, some existing protocols effectively collapse into the same BRAIN protocol. This observation will allow us to reduce the number of alternatives that need to be studied and to relate the choice to well-defined design decisions.

Among others, the targeted BRAIN protocol should have the following features:

?? support for paging to extend idle MN's battery lifetime;

?? efficient routing to find optimal routes in the BAN;

?? use of address aggregation to decrease the amount of per-host entries necessary;

?? seamless handover support to decrease service degradation of frequently moving MNs;

?? no single point of failure;

?? optimised MN-to-MN routing; and

?? independence of global MIP. The BRAIN protocol should not depend on any macro mobility protocol. (MIP is currently not the exclusive way to handle macro mobility problems other protocols e.g. SIP are existing)

Table A3-4 summarises the above principles for each protocol.

Let us first investigate the differences between **CIP** and **HAWAII**. While the two protocols are largely similar in philosophy, there are numerous differences in the implementation. The most apparent difference is that **CIP** uses data packets for paging and to update location information, while **HAWAII** uses explicit signalling messages. **CIP** nodes must snoop each packet to extract location information of MNs similar to Ethernet switches, while **HAWAII** routers can be legacy IP routers with enhanced control software. To allow easy design of nodes of a BRAIN network, **CIP** can be modified to use explicit signalling messages without changing the basic operation of the protocol.

Another difference between **HAWAII** and **CIP** lies in the routing of location update messages, which, in turn, determine the path taken by downlink data packets. **CIP** nodes address their location update packets to the gateway, and the resulting reverse downlink path will always be the shortest path from the gateway to the current BAR the MN is attached to. In contrast, **HAWAII** path set-up messages are addressed to the previous BAR. This may result in sub-optimal routing in **HAWAII** if the network topology is not a tree. Figure 3 shows migrations of a **HAWAII** mobile node. The blue arrows indicate the path taken by downlink data packets before the handover, while the red arrows correspond to the path taken by the path set-up message.

The first node that has a per-host entry for the migrating MN encountered by the path set-up message on its way toward the previous BAR will be the cross-over node. The part of the downlink path between the BMG and the cross-over node will remain untouched and will be the part of the new route as well. The new path is not the result of a shortest path calculation, but merely a concatenation of two independent segments. One is part of the old path from the BMG to the cross-over node, the other is the shortest path from the new BAR to the cross-over node. The combination of these two segments can lead to sub-optimal routing as shown in the next figure.

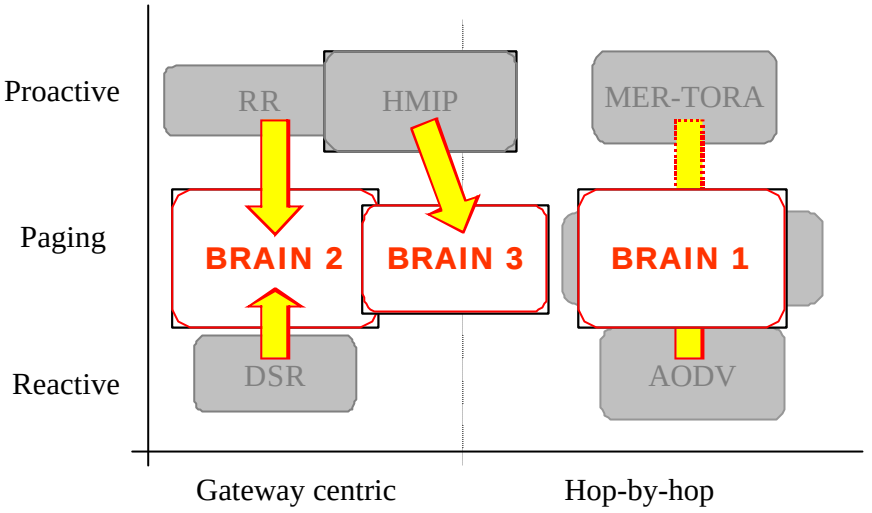


Figure A3-35: BRAIN Proposals

This shortcoming of **HAWAII** routing can be overcome by directing the update message toward the gateway instead of the previous BAR. The operation of this modified **HAWAII** protocol would be fundamentally identical to the operation of **CIP** with explicit signalling messages. We will term this envisioned common protocol **CIP-HAWAII**.

Let us now assume that a BRAIN access network is using **AODV** routing. Due to the fact, that the BRAIN access network is not a real ad-hoc network, each session starts either from the mobile host or from the gateway. On session start-up either the mobile host broadcasts a search for the gateway or vice-versa. In the former case the search will be very limited in scope, as the access points that first receive the search know the location of the gateway precisely. Therefore this case is not discussed further. In case of the gateway querying the MN the search message will be propagated in the entire network. When the MN is found at its current BAR a route reply message will be sent back to the gateway. As this message travels toward the gateway, it will configure the next hop information referring to the MN in each router along the way. The series of these next hop information elements constitute the route to the mobile node.

Since broadcast messaging is not acceptable for BRAIN, we need to limit the scope of the search performed by the gateway. That is, we have to introduce loose location tracking of mobile hosts that are not involved in communication. This can be achieved by maintaining proactive information about the MN in selected nodes of the BAN. Idle MNs occasionally report their whereabouts to these nodes in a proactive fashion. Such location updates can happen, for example, when the MN moves between administratively defined (paging) areas. In this case the broadcast search of the gateway can be by these nodes based on the proactive location information.

The operation of the **AODV** protocol with the above extension will become very similar to that of **CIP-HAWAII**. Upon session start-up a limited broadcast search message is used to find the host. The reply to this message builds up a per-host route to the mobile host in the form of next hop entries. This route is used to send packets to the host. Uplink control packets update this path.

The similarity of **CIP-HAWAII** and **AODV** shows a fundamental principle of micro mobility protocol design. Both fully proactive and reactive protocols have serious drawbacks. Proactive protocols require frequent location updates, while reactive protocols result in extensive searching. An intermediate solution is preferable that balances the two. The location of active nodes should be updated frequently to make communication possible. In contrast, the location of idle mobile nodes need not be tracked exactly, rather they need to be searched (paged). This intermediate solution can be reached from both proactive and reactive protocols by adding paging support or loose location tracking respectively. By adding loose location tracking to **AODV**, it naturally becomes similar to **CIP** and **HAWAII**, which are proactive protocols with paging support.

Let us now investigate, how the **DSR** protocol could be applied in a BRAIN access network. Due to the source routing nature of the protocol, the gateway has the full path to each mobile node the network serves, including the BAR. In the BRAIN scenario, it is unnecessary to include the full path in the source routing header. It is enough to place the address of the current BAR and use loose source routing. In this way only the BARs and the gateway need to run the **DSR** routing protocol, intermediate nodes may run any legacy routing protocol. The protocol operation then becomes very simple. The source routing option in uplink data packets updates the location information in the gateway. The gateway adds a loose source routing option to downlink data packets with the address of the current BAR. This network essentially represents a single level of hierarchy, as only the gateway maintains location information about mobile nodes.

The **DSR** protocol is a reactive protocol. Therefore, when location information in the gateway times out, it uses broadcast messages to find MNs. Similar to **AODV** the scope of such broadcast messaging should be limited. This is possible by introducing loose location tracking. MNs are required to occasionally report their location to the gateway. (E.g., when moving between administratively defined (paging) areas.) In addition, broadcast search operations are limited to the BARs belonging to the area the MN last reported from.

The **RR** proposal implements a single level of hierarchy similar to **DSR** applied to the BRAIN environment. However, **RR** is a fully proactive protocol, where MNs send regional registration messages on each move. To save power, paging areas similar to **CIP** or **HAWAII** can be defined, so idle mobile nodes have to report their location only when moving between such paging areas. This extension makes **RR** fundamentally similar to **DSR** in a BRAIN environment. Upon session start-up a limited broadcast search message is used to find the host. The reply to this message is sent directly from the BAR to the gateway, which learns the location of the MN from that message. This location is then used on tunnelling or source routing the downlink data packets addressed to the MNs. Note that the similarities between the extended **RR** and **DSR** protocols closely resemble to the similarities between the **CIP-HAWAII** and **AODV** protocols.

Address aggregation is a feature available only in the **MER-TORA** protocol, but is applicable to all hop-by-hop protocols. In **MER-TORA,** each BAR has a dedicated block of addresses assigned to it. A newly arriving MN can pick its IP address from the address block of that BAR where it has appeared. (This address can be used as either native or co-located care-of address.) This IP address is reachable by prefix based routing present in the network so the MN is addressable without per-host entries until it remains connected to its initial BAR. When the MN changes to a new BAR per-host entries will appear in a limited set of nodes. The further the MN moves from the initial BAR, the larger this set will be. Nodes high in the hierarchy will have per-host entries only for MNs far away from their initial BAR. In short, this solution decreases the number of needed per-host entries in the BAN and is beneficial in large networks.

Seamless handover support is present in all proposals, except **AODV** and **DSR**. However, using the ideas implemented in other proposals seamless handover support could be added to **AODV** and **DSR** as well. This task should be performed during the design of the BRAIN protocol design if one of these protocols is selected as the basis for the BRAIN protocol.

Each BRAIN access network that has only one gateway has an inherent single point of failure. Therefore the BRAIN protocol must allow support for multiple gateways. Moreover it must provide the protocol mechanisms that detect the loss of a gateway (or its connectivity to the Internet) and redirect traffic to other gateways. This is currently present only in the ad-hoc protocols, but appropriate extensions can solve the problem for other proposals as well. Having multiple gateways is beneficial in case of both gateway centric and hop-by-hop protocols. In this case, the networks

Finally, we note that the **RR, HMIP** and **HAWAII** protocols are coupled to Mobile IP and cannot work without it. However, none of them are transparent to Mobile IP, that is mobile nodes must implement additional micro mobility specific processing besides basic Mobile IP. Dependence on global mobility limits the applicability of the protocol, therefore we suggest that the final BRAIN protocol be independent, but interoperable with global Mobile IP.

A3.6.3.3 Possible "BRAIN" protocols

In the previous sections we have identified two important mobility protocol design decisions and have used these to classify existing micro mobility protocols. Next, we have listed a set of protocol features and have shown that all existing protocols support some of these features, but none of them supports everything. We have made suggestions on adding support for the missing features and have seen that by adding these features, we effectively carry ideas and solutions to each protocol from alternative protocols.

We claim that by applying the above mentioned changes the differences between particular protocols are becoming marginal (limited to e.g., message formats, etc) and protocol groups are being formed. In other words, by selecting one or another original micro mobility protocol we may arrive to fundamentally the same BRAIN protocol after adapting it to our requirements. We further claim that the differences that still remain between the protocol groups can be attributed to the design decisions that we used for our protocol classification.

The modifications applied to each protocol and the resulting grouping are shown in Figure A3-35. The figure shows that both proactive and reactive protocols must move toward the centre, that is, toward a combined approach. This is needed to avoid both frequent location update messages and broadcast search.

By using explicit signalling messages in **CIP**, correcting the sub-optimality in the routing of **HAWAII** and adding loose location tracking to **AODV** we essentially arrive to similar protocols. Any of these delivers unmodified IP packets in a hop-by-hop manner according to per-host location entries found in the nodes along the downlink path. Any of these allow multiple levels of hierarchy to be built. Location of idle nodes is tracked only approximately and searching (paging or route requests) are used to locate them.

The combined **CIP-AODV-HAWAII** protocol can be extended with features present in **MER-TORA** (also in the "hop-by-hop" category in Figure 1), such as address aggregation or the lack of single point of failure. This extended protocol would have most major advantages of **MER-TORA** but would still be simpler than **MER-TORA**, due to the lack of a complex, fully ad-hoc protocol, like TORA. Therefore, we suggest excluding MER-TORA and using the extended **CIP-AODV-HAWAII** protocol. This protocol is termed "**BRAIN 1"**, as shown in Figure A3-35.

The **BRAIN 1** protocol has the following properties.

- ?? Standard IP packet format (no tunnelling, no source routing option).
- ?? No central entity, the location database is distributed along the downlink path.
- ?? Optimal route selection between MN and the BMG.
- ?? All IP network nodes must be BRAIN aware.
- ?? Works without global Mobile IP.
- ?? Uses address aggregation to decrease the number of per-host entries.
- ?? Uses explicit signalling messages to update location information and for paging.
- ?? Location updates may be stopped at crossover routers and do not have to reach the BMG.
- ?? Paging support for idle nodes.
- ?? Seamless handover support.
- ?? No single point of failure.
- ?? Optimised MN-to-MN routing.

This protocol group suits large BANs, where a single gateway could not process the mobility related signalling of all nodes. It uses unmodified IP packets. Its major disadvantage is that all nodes in the network must be BRAIN aware, thus the protocol cannot be applied to an existing IP network without upgrading most of the equipment.

Another group of proposals leading to similar protocols consists of **RR** and **DSR**. The combined and enhanced **RR-DSR** protocol is gateway centric and keeps location only in the gateway. All location updates are sent to the gateway, which uses tunnelling or loose source routing options to transmit packets to the appropriate BAR over a routed IP network. On Figure A3-35 4 this protocol is termed "**BRAIN 2."**

The **BRAIN 2** protocol has the following properties.

- ?? Packets are tunnelled or contain source route option.
- ?? Location of MNs is tracked by a single entity.
- ?? Only a minor part of the nodes should be BRAIN specific. Packets are forwarded using standard IP protocols.
- ?? Works without global Mobile IP.
- ?? Paging support for idle nodes.

?? Seamless handover support.

?? Multiple gateway support, including mechanisms to ensure resilience to gateway failures.

This protocol group suits a smaller BAN where a single entity is enough to keep track of all mobile nodes. This prevents the optimisation of MN-to-MN routing; all such packets have to go through the gateway. The major advantage of the protocol is the ability to run over legacy IP networks. (Optionally source route support is needed.) The gateway and BARs can be connected to an existing network and are readily operational. Naturally, it is possible to include more than one gateway into a **BRAIN 2** network both to increase robustness and performance.

The **HMIP** protocol combines elements from both protocol groups. First, it is similar to **BRAIN 1** protocol in that it supports multiple levels of hierarchies. It can benefit from address aggregation and can limit the signalling load on the gateway by stopping location updates in the crossover node. Second it is similar to **BRAIN 2** protocols, in that it uses tunnelling. In fact, using the **HMIP** protocol with only one level of hierarchy we can come back to **BRAIN 2**, more specifically to **RR**. By adding the appropriate functions to **MAILINEN**, an in-between protocol, **BRAIN 3** could be developed besides **BRAIN 1** and **BRAIN 2**.

The **BRAIN 3** protocol has the following properties.

?? Packets are tunnelled or contain source route option.

?? No central entity, the location database is distributed along the downlink path.

?? (Near) optimal route selection between MN and the BMG.

?? There might be BRAIN unaware IP routers in the network.

?? Works without global Mobile IP.

?? Uses address aggregation to decrease the number of per-host tunnelling entries.

?? Location updates may be stopped at crossover routers and do not have to reach the BMG.

?? Paging support for idle nodes.

?? Seamless handover support.

?? No single point of failure.

?? (Partly) optimised MN-to-MN routing.

The major advantage of **BRAIN 3** is in the deployment. Mobility can be added to a small existing network by implementing only one level of hierarchy. This would essentially be a **BRAIN 2** network. Then later, as the network grows, more levels may be added to enjoy the benefits of multiple levels, such as decreased signalling load and tunnelling entries. The resulting network, however, will work only with tunnelling.

A3.6.4 Micro mobility and QoS

In this section we propose a QoS architecture framework as the basis for the BRAIN QoS architecture. We give some overview of the proposed architecture and try to evaluate the proposed architecture and the mobility aspects. Finally, we give some thoughts about the overall operation of this architecture and the inter-working with micro mobility schemes.

Looking at the different types of applications for multimedia transfers, we can identify two main types: applications that are aware of their exact resource needs and applications, which do not expect special treatment from the network, even though these applications would benefit from better than best-effort QoS. An example of the former would be a RSVP-aware application, and an RTP-aware application would fit the latter type. Both applications should be supported by a QoS architecture. The issues raised by the Internet Architecture Board ([A3.64]) and general user requirements have also driven the design of this architecture.

A3.6.4.1 The framework

The IETF Integrated Services over Specific Link Layers working group (ISSLL) has come up with a framework for sending RSVP-controlled traffic ([A3.46], [A3.53], [A3.76]) through Differentiated Services (DiffServ, [A3.42], [A3.51]) networks while giving QoS assurances to user flows ([A3.48]). This framework provides powerful mechanisms for allowing both per-application resource

requests through RSVP but aggregation of flows within the DiffServ network. Resource states, or knowledge of resource availability, need only be kept at network edge routers and a central Bandwidth Broker (BB).

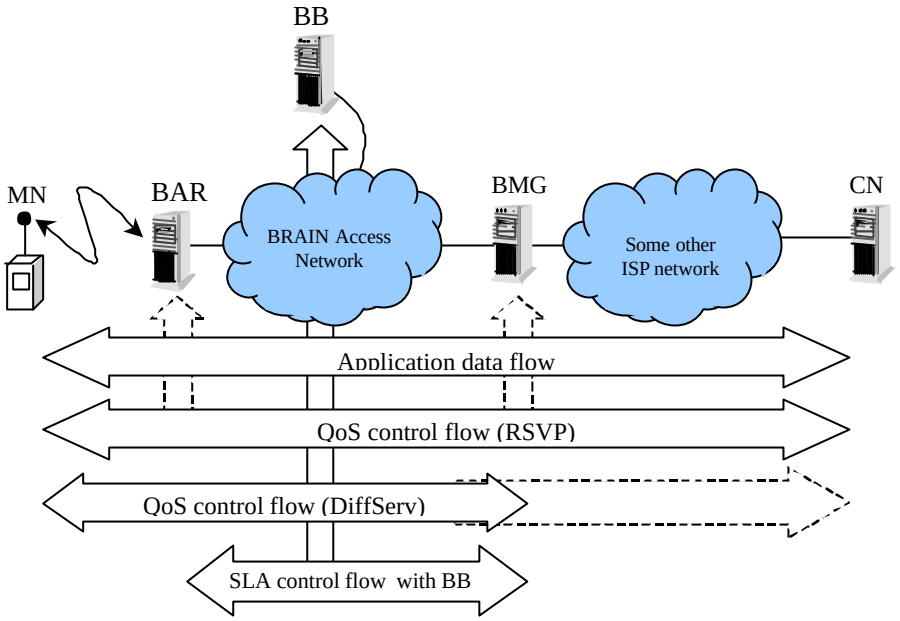


Figure A3-36: The QoS-aware Nodes and QoS-signalling

In addition to the per-application signalled reservation, the presented architecture also allows direct DiffServ Code Point (DSCP) marking for applications that are not QoS and RSVP aware, but the user would want to receive some better service for these data exchanges, the RTP-based VoIP flows, for example. The direct marking can indicate some relative priority, for example, Assured Forwarding (AF, [A3.49]) classes, but nothing forbids the network operator to provide direct DSCP values for predefined services, like values for 64, 128, and 256 kbit per second low latency assured bandwidths.

In a way, for unicast flows, the RSVP-signalled transfers get a circuit-switched (CS) connection, while the direct-mapped flows get a priority-enhanced packet-switched (PS) service. In a CS connection, there is always some connection setting up phase prior to the data transfer. With RSVP, the default virtual CS connection is however unidirectional, two set up phases are required for a duplex connection. The direct-mapped transfer is less reliable due to the lack of setting up a dedicated reservation, but faster to initiate.

From these viewpoints the issue of micro mobility and QoS interoperability is fundamentally a question of providing mobile CS connections, the PS connections should then provide no further problems.

The BAR component is the first IP-based node to which a flow originated by a MN arrives to (or the last IP-based node before the MN for downlink transfers). A BAR can be the common first IP-based node for several base stations, and is therefore in charge of resource co-ordination for the base stations under it. The BAR could have an interface for each of its base station, and could do per-base station resource allocation – for example, to the BAR, a base station is just one of its interfaces on the downlink direction.

The BAR node has much of the same functionality as a DiffServ ingress node upgraded with functionality needed to support RSVP signalling and Service Level Agreement (SLA) management. A DiffServ ingress node is in charge of admission control, service negotiation, and adding the proper DiffServ Code Point (DSCP) to the IP header in order to produce proper forwarding behaviour in network routers. The mapping is based on the SLAs negotiated between the subscriber and the ISP and can be varying according to the present time and date and network load. The BAR gets the information for the SLAs from the Bandwidth Broker using the Common Open Policy Service (COPS, [A3.59], [A3.55]) protocol. The SLAs can be changed dynamically.

The Brain Mobility Gateway (BMG) has much the same functionality as the BAR, but for flows arriving from the external networks. Admission control functionality controls the load arriving from the external ISPs, dropping or remarking packets if they do not conform to the SLAs or Tspec for the mobile being reached (for which the information is updated by the BB and RSVP-reservations).

The internal **routers** forward packets according to the destination and the Behaviour Aggregate (BA). DiffServ policing and shaping can be performed if the load exceeds the amount of resources allocated to

Type	*Originator*	*Availability*
RSVP-based	Mobile Node	Yes, enhanced with the DCLASS object
RSVP-based	Correspondent Node	Yes, the BMG does the classification
Non-RSVP-based	Mobile Node	Yes, through direct DSCP marking
Non-RSVP-based	Correspondent Node	No, MN would need to provide the BMG with some indications of QoS, possibly with the same DCLASS object

Table A3-5: Forwarding Assurances for Flows

different PHB aggregates. By default this should only happen to best-effort traffic if the BAR and BMG do a proper shaping of flows admitted into the network.

In order to allow for accurate resources utilisation and better overall service, the AR and MG nodes need to keep state of the overall resources availability and the utilisation of the different forwarding classes. A central Bandwidth Broker (BB) is in charge of the overall admission control and service provision. The BB stores, and propagates to the BAR and BMG nodes, the ISPs policies on admission control (who, when, what service, etc.). Depending on the usage model, the SLA management can be handled by periodic broadcasts of the available SLAs to the network nodes or the BAR/BMG can ask for the admission control decisions each time there is an incoming explicit QoS request or simply a new flow, either from the inner or the outer network ([A3.71]). Either way, the SLA management can be very dynamic. The BB could also store, or request from the home network, the SLA information of each visiting MN.

By default the Mobile Node (MN) interacts with the QoS-structure of the BRAIN network with RSVP. The set of allowed QoS parameters must be quite limited to allow for more simple and direct mapping of RSVP to DiffServ behaviour aggregates. Additionally, the MN must be configured to understand what types of service it can request through direct DSCP marking in the IP packets and what are the charging issues of those services.

A key point in forwarding RSVP-reserved flows is how to map the RSVP reservations into proper DiffServ forwarding aggregates. The natural way would be to use the EF behaviour aggregate ([A3.65]), since the application signalled a specific request with probably bandwidth and delay parameters. The mapping is however not only based on the Per-Hop behaviours (PHB) but more like on Per-Domain behaviours (PDB, [A3.72],[A3.66]). The forwarding treatment must provide a constant service across the whole domain, thus the forwarding treatment of PHB dedicated to RSVP must take into account the whole path between the BAR and BMG nodes.

An important decision in this architecture would be to leave the flow marking to the MN. This has two main advantages. First it takes part of the packet handling from the BAR away, since the BAR does not need to check for the IP- and transport headers for information for multifield flow classification, but can rather rely on BA classification. Second, it allows the MN to use IPSec payload encryption (ESP, [A3.67]) (and any IP-within-IP tunnelling). If the payload was encrypted, the BAR might not have enough information to do multifield classification.

The RSVP DCLASS object ([A3.47], [A3.73]) which allows an RSVP message to include information about the suitable DSCP for this flow, can provide for some help in this. When the MN requests some defined resources for an uplink transfer, the BAR node could add a DCLASS object to the returning RESV-message. When the MN receives the RESV with the DCLASS object, it would use the presented DSCP for this transfer.

The use of the RSVP DCLASS object is only applicable for uplink transfers, since the CN may not be able to handle the DCLASS object. Even if the DCLASS object is used end-to-end the proposed DSCP may not persist all the way from the CN to the BMG. Therefore, the BMG must do the multifield classification for downlink packets as well as it can, even in the presence of IP-tunnelling.

For non-RSVP application, a QoS control tool could allow the user to request better-than-best-effort service to certain flows by marking packets of chosen flows with proper DSCPs. The triggered services per DSCP could be configured through the normal MN dynamic configuration when entering a domain. These still leaves out non-RSVP based flows arriving to the MN. The MN would also need some mechanism to trigger proper DSCP marking and PHB at the BMG. In case of resources outage in a certain non-RSVP service class, the MN would need to be informed about the congestion. This could be done with an ICMP message, preferable during the admission control phase before a new flow is even admitted into the network.

Type	*QoS flows involved*	*Description*
Intra-BAR	Both RSVP and non RSVP flows	Handover within the same BAR, so the routing does not change. Same to all flows.
Inter-BAR	Both RSVP and non RSVP flows	Handover between two BARs, where the BMG remains the same. Affects RSVP-flows a fraction more than non-RSVP flows.
Inter-BMG	Both RSVP and non RSVP flows	Handover, where the BMG changes for some flows. This is the most troublesome handover, since the most control signalling is needed, possible even end-to-end. Greatly affects the RSVP-flows.

Table A3-6: HandoverTypes for Independent Flows

A further benefit of having the MN mark by itself the IP packets with some DSCP values is that these values could also be used for link layer scheduling purposes. The protocol stack below IP is in charge of scheduling the proper packets to be sent on the wireless link. By having a field in the IP-header to mark a priority of each packet, the link layer scheduler could more straightforwardly and effectively deliver IP packets in the right order to the wireless interface for transmission.

A question can rise as to how the mobile can be trusted to do the marking by itself. If we add charging issues within the context of QoS, we can allow the MN to do the marking. If the MN tries to trigger better service, or to send more data, than it has requested resources for, the DiffServ flow shaping will drop or remark the packets which do not conform to the indicated traffic specification ([A3.43]), or the MN user will be charged more due to better service.

Another central implementation issue is the co-ordination of radio resources. The availability of radio resources in different cells and the admission control to these resources is a difficult inter-layer communication issue. From the point of view of a BAR, and IP, an interface to a cell can be though to be a "single link" with certain resource availability. This abstraction can simplify the design on the QoS entities and their interworking. However, the link and physical layers see the (radio) resource availability and co-ordination totally differently. Problems will arise when the capacity of a given cell changes, as for example, in wide band CDMA, due to the fading and distance between MNs and the base station antenna. These changes would be needed on the IP layer to properly calculate the capacity available to MNs. These issues, among others, are tackled by the IP2W (IP to Wireless) group of the project ([A3.58], [A3.69]).

A3.6.4.2 Handovers and mobility

Handovers in this architecture can be divided into two classes, whether the resources were reserved with RSVP or not. When the MN moves, depending on the implementation, the MN or network needs to find a new cell, which can support

1. the radio resource context of the MN, in terms of DSCPs and PHBs from which some may be part of RSVP reservations,
2. access network resources for the same marked flows, and
3. the same resources on the link from the BMG to the external network.

The different handover scenarios can be further divided depending on whether the BAR or the BMG nodes change. If a suitable cell is found and the MN does a handover, the routing of packets to and from the MN change. For the QoS-query during handovers, we need to define an internal signalling protocol, which allows the MN or network to query and reserve resources from all three logical areas mentioned above. This mechanism could be based on both RSVP and COPS, for example.

All in all, we can identify six types of handover situations, which create different amounts of control signalling between different entities. The same physical handover can create different logical handover situations to different flows (Table A3-6).

If the BAR node does not change during a handover, the handover control therefore only need to check for radio resource availability, since the flows will still use the same routing paths between the BAR and BMG nodes. Even the admission control part may be left out, since the admission control has already

been done for this BAR when the MN initiated the transfers. This applies for both RSVP-signalled and non RSVP-signalled flows.

If for a given RSVP-reserved uplink flow the BAR changes, but the BMG stays the same due to similar routing, the handover control need to check the radio resource availability and the access network resources from the central bandwidth broker. Also, the BAR may need to check for admission control at the same time. This type of handovers need, together with other information, some knowledge of the old and new BARs routing structure, in order to limit the handover related signalling.

With RSVP, the MN needs to send periodic RESV or PATH messages for each flow, depending on whether the MN is a receiver or sender, to refresh the end-to-end reservations. In order to enable a shorter period where there is no reservation, the refresh messages should be sent immediately after a handover to trigger at the BMG an update of the MN's location.

The resource co-ordination due to mobility is much more affected by the change of BMGs. If the BMG changes, the MN may either need to re-reserve the resources end-to-end or wait until a scheduled refresh message reaches a router which has the state of the reservation, and can thus initiate an update in the reservation states on the path.

If the BMG changes, resource co-ordination and allocation become more complex, since prior to the handover the handover control need to check for resource availability on all three links, the radio link, the access network (which is seen as a single link) and the network edge link. Therefore having a steady routing path between the mobile's location and the external network is important. More specifically we would need to keep the same BMG during handovers if possible. However, due to scalability considerations, the BMG should not be tied to MN on a permanent basis.

For flows that have initiated without RSVP, the handover control and subsequent resource signalling is similar to the RSVP-reserved flows. The difference is that there should be no "empty period" during which the BAR/BMG nodes refresh their RSVP-related routing tables and resource allocations. The MN should therefore get a more straightforward service for these flows. The same resource availability signalling is required for checking the resources on the mentioned three links. If resources are available, the BB, BAR and BMG only need to update their resource states. If the BMG changes the resource allocation is also faster since an end-to-end RSVP signalling is not needed.

If resources are not available, the MN will need to be signalled about the condition. With RSVP reserved flows, the RSVP error reporting applies directly and with the direct-mapped flows, the MN could get an ICMP error message. It should be noted that, in order to minimise the needed error signalling, the network should perform the resource negotiations with the BB prior to the handover in order to check beforehand that resources are available in the new cell after the handover.

Assuring the resources when a MN changes cells is also one of the key problem areas. Several research projects have discussed making advance reservations in neighbouring cells ([A3.56], [A3.75]). These resources mainly cover the radio resources, but we would also take into account the resources of the access network between the network edges and the resources of the interface between our domain and the next operator.

The MN could make tentative reservations in neighbouring cells and actually take the resources if it moves into the neighbouring cell. When several MNs make those tentative reservations, resources can be unnecessarily removed from actual use. The way these pre-reserved resources are used is an important question. For example, allocating the tentative resources for best-effort traffic, while the roaming MNs are making use of them is not a good solution. Macro-diversity, sending or receiving data through two separate base stations needs also be studied.

A3.6.4.3 Discussion

The presented architecture framework is, as we see it, very much independent of the underlying micro mobility scheme. By using aggregate packet forwarding and no explicit reservations in intermediate routers we can provide the most flexible base, from the point of view of QoS, for micro mobility techniques.

The DiffServ QoS architecture, which constitutes the core of our access network, is still very much affected by the operation of the chosen micro mobility protocol, not taking into account whether it is one of the proposed schemes or a totally new one. Especially the following design decisions need to be considered:

?? Use of tunnelling,

?? changing the end-point IP-address the CN will use for communications with the MN during the lifetime of a connection,

?? multicasting packets to several base stations consumes resources,

?? having a fixed route to the outer network (always through the same BMG), and

?? adaptability and techniques (speed and reliability) to changing routing paths.

Using only BA classification in intermediate routing can alleviate tunnelling considerations. This requires the DSCP to be copied from the inner IP-header to the outer one. DiffServ operation with tunnels is discussed in **(**A3.51]). Multicast considerations with DiffServ are discussed in [A3.42] and [A3.54]. Keeping a constant IP address for a MN and the adaptation to routing changes are implementation specific and not related to the DiffServ architecture itself. However, as Figure A3-34 clearly presented, the ability of the micro mobility protocol to track the new location of a MN can create very problematic situations in view of QoS resource allocation. The route between the BMG and BAR may not be based on the shortest path, but rather the packets will flow through several needless routers.

Other less mobility-related advantages of this architecture include a broad and flexible support of different QoS needs, both the application-driven RSVP and a transport layer option as DSCP marking. Scalability is not hindered even though RSVP is used, since the access network is still purely DiffServ-based and the RSVP-like states are kept only at the edges of the network. Resource co-ordination allocation can also be performed efficiently, and use of open IETF protocols leaves much room for concentrating on studying different implementation issues.

Security issues, among others, related to the proposed framework still need to be evaluated. One key issue, which also requires some studying, is the possibility to perform Constraint Based Routing ([A3.68]), routing that is based on resource availability, allowing to route flows between the same network edges through different paths. A requirement is however that flows enter and emerge from the same network edge nodes, but the internal routing path between these nodes could be dynamically varying according to resource availability. This helps both the MN's user in getting better overall service and the network operator in having a more even utilisation of the network.

A3.6.5 Conclusion

In the MM part of the paper we selected key properties of existing micro mobility protocols. Based on these properties we grouped the protocols into categories. By identifying required features of existing micro mobility protocols, we arrived to three protocol outlines that can form the base of the future BRAIN protocol.

The **BRAIN 1** protocol is a multilevel protocol with distributed location tracking that uses standard IP packets without tunnelling or source routing.

The **BRAIN 2** protocol is a single-level protocol in which a central gateway keeps track of the mobile nodes and uses tunnelling or source routing to deliver packets to the actual point of attachment.

The **BRAIN 3** protocol is a hybrid that uses multiple levels of hierarchy and tunnelling at the same time.

All three protocol types above represent a trade-off between desirable protocol properties, such as easy deployment, scalability, ability to support QoS, etc. The selection of the appropriate protocol type for use in a BRAIN Access Network highly depends on the exact scenario and on the preferred trade-off.

In the QoS part of this paper the presented architecture provides both direct (RSVP) and indirect (DiffServ) QoS reservations in order to support different applications. It is a scalable architecture and provides end-to-end guarantees, together with guarantees within a single BAN, between two MNs. The use of DiffServ in the access network should leave most room for independent evolution of micro mobility schemes, although some micro mobility design issues, that were raised, can have a negative effect on the overall QoS performance. The presented architecture only requires a new protocol for handling the QoS queries during handover, other protocols needed are already defined by the IETF.

A3.6.6 Paper References

[A3.41] Charles E. Perkins, Elizabeth M. Royer, Samir R. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing" Internet Draft, draft-ietf-manet-aodv-06.txt, July 2000.

[A3.42] Blake, S., Black, D., Carlson, M., Davies, E., Wang,, Z., Weiss, W., "An Architecture for Differentiated Services". Internet Engineering Task Force, Request for Comments (RFC) RFC 2475, Dec. 1998.

[A3.43] Bernet, Y., Blake, S., Grossman, D., Smith, A., "An Informal Management Model for DiffServ Routers". Internet Engineering Task Force, Internet Draft, July 2000 (draft-ietf-diffserv-model-04.txt).

[A3.44] Boyle, J., Cohen, R., Durham, D., Herzog, S., Rajan, R., Sastry, A., "The COPS (Common Open Policy Service) Protocol". RFC 2748, January 2000.

[A3.45] Boyle, J., Cohen, R., Durham, D., Herzog, S., Rajan, R. and Sastry, A., "COPS Usage for RSVP". Internet Engineering Task Force, Request for Comments (RFC) 2749, January 2000.

[A3.46] Braden, R., Clark, D. Shenker, S., "Integrated Services in the Internet Architecture: an Overview ". Internet Engineering Task Force, Request for Comments (RFC) 1633, June1994.

[A3.47] Bernet, Y., "Format of the RSVP DCLASS Object". Internet Engineering Task Force, Internet Draft, October 1999 (draft-ietf-issll-dclass-01.txt).

[A3.48] Bernet, Y. at al, "A Framework For Integrated Services Operation Over Diffserv Networks". Internet Engineering Task Force, Internet Draft, May 2000 (draft-ietf-issll-diffserv-rsvp-05.txt, expires November 2000).

[A3.49] Baker, F., Heinänen, J., Weiss, W., Wroclawski, J., "Assured Forwarding PHB Group". Internet Engineering Task Force, Request for Comments (RFC) 2597, June 1999.

[A3.50] Baker, F., Iturralde, C., Le faucheur, F., Davie, B., "Aggregation of RSVP for IPv4 and IPv6 Reservations". Internet Engineering Task Force, Internet Draft, March 2000 (draft-ietf-issll-rsvp-aggr-02.txt).

[A3.51] Blake, S., et al., "An Architecture for Differentiated Services". Internet Engineering Task Force, Request for Comments (RFC) 2475, December 1998.

[A3.52] Black, D., "Differentiated Services and Tunnels". Internet Engineering Task Force, Request for Comments (RFC) 2983, October 2000.

[A3.53] Braden, R., Zhang, L., Berson, S., Herzog, S., Jamin, S., "Resource ReSerVation Protocol (RSVP) -- Version 1, Functional Specification". Internet Engineering Task Force, Request for Comments (RFC) 2205, September 1997.

[A3.54] Bless, R., Wehrle, K., "IP Multicast in Differentiated Services Networks". Internet Engineering Task Force, Internet Draft, September 1999 (now expired) (draft-bless-diffserv-multicast-00.txt).

[A3.55] Chan, K., et al., "COPS Usage for policy Provisioning (COPR-PR)". Internet Engineering Task Force , Internet Draft, October 2000 (draft-ietf-rap-pr-05.txt).

[A3.56] Chen, J-C, et al. "QoS Architecture Based on Differentiated Services for Next Generation Wireless IP Networks". Internet Engineering Task Force, Internet Draft, July 2000 (draft-itsumo-wireless-diffserv-00.txt).

[A3.57] Cambell, A. T., Kim, S., Gomez, J.,Wan, GY., Turanyi, Z., Valko, V., "draft-ietf-mobileip-cellularip-00.txt", IETF mobile IP Working Group Document, December 1999.

[A3.58] BRAIN project deliverable 2.1 "BRAIN Access network requirements, specifications and evaluation of current architectures and technologies and their requirements: core network and air interface", September 2000.

[A3.59] Durham, D., et al. "The COPS (Common Open Policy Service) Protocol". Internet Engineering Task Force, Request for Comments (RFC) 2748, January 2000.

[A3.60] A. O'Neill, G. Tsirtsis, S. Corson, "Edge Mobility Architecture" Internet Draft, (work in progress), draft-oneill-ema-02.txt, July 2000.

[A3.61] Fankhauser, G., Hadjiefthymiades, S., Nikaein, N., Stacey, L., "RSVP Support for Mobile IP version 6 in Wireless environments". Internet Engineering Task Force, Internet Draft, May 1999 (draft-fhns-rsvp-support-00.txt).

[A3.62] Ramjee, R., La Porta, T., Thuel, S., Varadhan, K., "IP micro-mobility support using HAWAII", Internet Draft, (work in progress), draft-ietf-mobileip-hawaii-00, June '99.

[A3.63] Ramjee, R., La Porta, T., Li, L., "Paging support for IP mobility using HAWAII", Internet Draft (work in progress), draft-ietf-mobileip-paging-hawaii-00.txt, June '99.

[A3.64] Huston, G. ”Next Steps for the IP QoS Architecture”. Internet Architecture Board, Internet Draft, August 2000, (draft-iab-qos-02.txt).

[A3.65] Jacobson, V., Nichols, K., Poduri, K., "An Expedited Forwarding PHB". Internet Engineering Task Force, Request for Comments (RFC) 2598, June 1999.

[A3.66] Jacobson, V., Nichols, K., Poduri, K., "The “Virtual Wire' Behaviour Aggregate". Internet Engineering Task Force, Internet Draft, March 2000 (draft-ietf-diffserv-ba-vw-00.txt).

[A3.67] Kent, S., Atkinson, R., “IP Encapsulating Security Payload (ESP)”. Internet Engineering Task Force, Request for Comments (RFC) 2406, November 1998.

[A3.68] Kompella, K., Awduche, D., "Notes on Path Computation in Constraint-Based Routing". Internet Engineering Task Force, Internet Draft, July 2000 (draft-kompella-te-pathcomp-00.txt).

[A3.69] Laukkanen, A., Bertin, P., Liljeberg, M., Suihko, T., “IP to Wireless Convergence Layer”. BRAIN Workshop, London, November 2000.

[A3.70] Malinen, J., Perkins, C., “Mobile IPv6 Regional Registrations”, Internet Draft (work in progress), draft-malinen-mobileip-regreg6-00.txt, July 2000.

[A3.71] Mameli, R., Salsano, S., “Integrated services over DiffServ network using COPS-ODRA”. Internet Engineering Task Force, Internet Draft, February 2000 (draft-mameli-issll-is-ds-cops-00.txt).

[A3.72] Nichols, K., Carpenter, B., "Definition of Differentiated Services Per Domain Behaviors and Rules for their Specification". Internet Engineering Task Force, Internet Draft, October 2000,draft-ietf-diffserv-pdb-def-01).

[A3.73] Syed, H., “Capability Negotiation: The RSVP CAP Object”. Internet Engineering Task Force, Internet Draft, September 2000 (draft-ietf-issll-rsvp-cap-00.txt).

[A3.74] Park, V., Corson, S., “Temporally-Ordered Routing Algorithm (TORA) Version 1 Functional Specification”, Internet Draft (work in progress) , draft-ietf-manet-tora-spec-02.txt, October ‘99.

[A3.75] Talukdar, A., Badrinath, B., Acharya, A., “MRSVP: A Resource Reservation Protocol for an Integrated Services Packet Network with Mobile Hosts”. In Proceedings of ACTS Mobile Summit’98, June 1998.

[A3.76] Wroclawski, J., " The Use of RSVP with IETF Integrated Services". Internet Engineering Task Force, Request for Comments (RFC) 2210, September 1997.

A3.7 A Framework for the Evaluation of IP Mobility Protocols (PIMRC paper)

This section presents a paper that was written by Philip Eardley, Andrej Mihailovic and Tapio Suihko and presented at PIMRC 2000

A3.7.1 Abstract

In this paper we suggest a classification scheme for IP mobility protocols and propose a framework for comparing them. We then use the framework to provide an initial comparison of recent proposals for supporting micro-mobility. The authors are part of Project BRAIN (Broadband Radio Access for IP based Networks) – a European collaborative project under the IST (Information Societies Technology) programme. One aim of the project is to propose an open architecture for wireless broadband Internet access, concentrating on issues in the access network.

A3.7.2 Introduction

The interest of this paper is host mobility (also known as terminal mobility) in an IP network.

The principal problem that mobility presents to a network is:- when a mobile host[43] (MH) moves onto a new base station (BS), how do we route packets to its new destination? We would like a solution that also (amongst other things) ensures that:-

?? the break in communications during the handover is as short as possible and that no (or only a few) packets are lost. Hence all applications, including real-time ones, will be supported.

?? The overheads from the messaging to achieve the re-routing are as low as possible. Included here is minimising the signalling load and latency, and also the storage and processing requirements at each router.

?? The solution is scalable, e.g. we can apply it whether we have a small or large number of MHs.

?? The solution is compatible with other Internet protocols, e.g. it does not interact adversely with Quality of Service (QoS) protocols.

Solutions to the basic mobility problem involve establishing some sort of dynamic mapping between the MH's identifier and its location (i.e. what the correspondent host (CH) wants to talk to vs. how to route packets through the network between the CH and MH). The best known proposal is Mobile IP (MIP) [A3.77], which solves the problem through using two IP addresses per MH – one acts as its permanent identifier, whilst the other acts as its temporary routable address (termed the Care-of-address, CoA) and the mapping between the two is stored at its Home Agent (HA). However, MIP[44] is a long way from the ideal solution outlined earlier, for example:-

?? Handovers may not be fast and smooth, because the MH must signal its change of CoA to the HA. This may take a long time if the HA is far away, perhaps in a different country.

?? The messaging overhead may be significant particularly if the HA is distant, as this will induce signalling load in the core of the Internet

?? MIP may interact with QoS protocols (DiffServ, IntServ), so making QoS implementation problematic. For example, MIP utilises tunnels and so packet headers – which may contain QoS information – become invisible.

However, MIP is relatively simple and robust and is likely to be ubiquitous. It thus appears to be a good way of handling global mobility and mobility between different operators. Meanwhile, more optimised solutions can be developed for regional[45] mobility. These exploit the significant 'localisation' of a MH's movement - typically, route updates travel to the nearest cross-over router[46] (as opposed to MIP where the HA is informed), thus reducing the signalling load in the core of the network and improving the re-

[43] also called a mobile terminal or node

[44] Some (but not all) of the problems are reduced by Route Optimisation of Mobile IP; there is not space to discuss it here.

[45] We use the rather vague term 'regional mobility', since an IP regional mobility protocol could (depending on its scalability) be suitable for a single IP domain up to the whole of an Autonomous System.

[46] Ie the last one common to the route from the CH to the old BS and the route from the CH to the new BS.

routing latency. Our overall solution therefore consists of MIP, to handle global[47] mobility, bolted on to a specialised regional[48] mobility scheme (Figure A3-37). The latter are the concern of this paper.

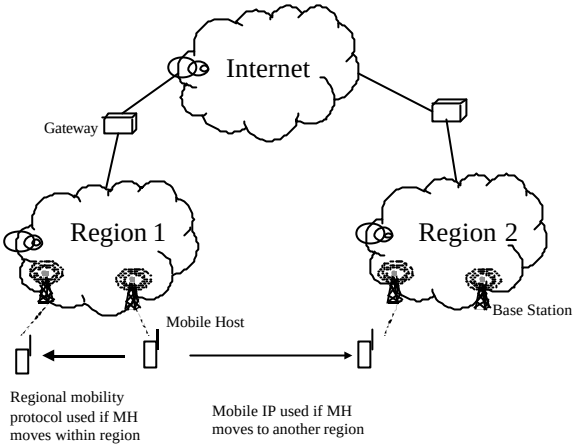


Figure A3-37: Global vs. Regional Mobility Management

In this paper we compare various regional mobility proposals (A3.7.5). In order to make this analysis more effective, we have developed an Evaluation Framework (A3.7.4) – which formalises the functions a protocol must do and what criteria to use to assess how well it does them. We also believe that the evaluation will be more effective if we first classify different IP mobility protocols (A3.7.3), to make more sense of the already very extensive research on IP mobility. The eventual objective of our work in the BRAIN project is to contribute improved IP regional mobility protocols. However, in this paper our main aim is to present our Evaluation Framework. In order to show that the Evaluation Framework can be useful (e.g. to identify key differences between protocols), we also present a preliminary application of it.

A3.7.3 Classification of IP Mobility protocols

The two major categories of *Regional Mobility* protocols are:

??***Proxy-Agent Architectures (PAA)***

??***Localised Enhanced-Routing Schemes (LERS)***

A Proxy Agents Architecture Schemes (PAA)

These schemes extend the idea of Mobile IP into a hierarchy of Mobility Agents (which are extensions of MIP's Foreign Agents (FAs) and/or HAs). A MH registers with its local Agent ('a') at the bottom level of the hierarchy ("MH is at Care-of-Address (CoA)"), which in turn registers with its nearest Agent at the next hierarchy-level ("MH is at Agent a"), and so on up the hierarchy towards the HA. This way, when the MH changes its CoA, the registration request does not have to travel up to the HA but remains 'regionalised'. Packets from a CH travel down the hierarchy, being tunnelled from one level to the next.

Examples include the initial Hierarchical Mobile IP [A3.81] and its alternatives, which place and interconnect Mobility Agents more efficiently: Mobile IP Regional Registration [A3.82], Transparent Hierarchical Mobility Agents (THEMA) [A3.83], Fast Handoff Methods [A3.84] and Hierarchical Mobile IPv6 [A3.85].

B Localised Enhanced-Routing Schemes (LERS)

These schemes introduce a new, dynamic Layer 3 routing protocol in a 'localised' area. There are several distinctive approaches:

B1 - Per host Forwarding Schemes: Inside a domain, a specialised path set-up protocol is used to install soft-state host-specific forwarding entries for each MH. The domain, which appears as a subnet to routers outside the domain, is connected to the Internet via a special gateway, which must be pointed to by the default gateway of the routers (or packet forwarding nodes) inside the domain. Examples include Cellular IP [A3.87] and Handoff-Aware Wireless Access Internet Infrastructure (HAWAII) [A3.88],[A3.89].

[47] There are non-Layer 3 solutions for "global" mobility, eg SIP [4], dynamic DNS.

[48] There are also Layer 2 solutions for "local" mobility, eg IAPP, L2TP.

B2 - Multicast-based Schemes: Multicast protocols are designed to support point-to-multipoint connections. So they share with IP mobility the same design goals of location independent addressing and routing and thus multicast-based mobility solutions have been proposed. A multicast-CoA is assigned to a single MH which can then be used to instruct neighbouring multicast-enabled routers to join the MH's virtual multicast group, either prior to or during handovers. This can be visualised as a multicast cloud centred on the MH's current location but also covering where it may move to. Examples include Dense mode multicast-based [A3.91],[A3.92],[A3.93] and the recent Sparse-mode multicast-based [A3.90].

B3 - MANET-based Schemes: MANET protocols were originally designed for Mobile Adhoc NETworks, where both hosts and routers are mobile, i.e. there is no fixed infrastructure. The routing is multi-hop and adapts as the MHs move and connectivity in the network changes. MANET protocols can be modified for our scenario, where there is a fixed infrastructure and only hosts can be mobile. Currently there is only one proposal in this category: MER-TORA [A3.94].

Figure A3-38 2 shows some of the many IP mobility protocols, which category they fall into and very roughly how they relate to each other.

A3.7.4 Evaluation Framework

In the Introduction, we listed some of the features we would like an IP mobility solution to have. In this Section, we expand on these and break them down into a more formal structure:- an Evaluation Framework. It has two dimensions:

A. Protocol Design Issues – the functional requirements for any IP-mobility protocol

B. Evaluation criteria – against which the effectiveness of a particular Solution to the Issues can be assessed.

In other words, we first decide what things a protocol must be able to do, and then how to assess how well the protocol does them.

A3.7.4.1 Protocol Design Issues

Here we list the Protocol Design Issues, along with a short explanation / discussion of each.

Packet Forwarding: Packet forwarding refers to the delivery of packets to and from the MH. In the 'traditional' Internet, this is based on shortest path routing (e.g. OSPF), where the aggregation of addresses means that routing can be prefix-based. However, this must be modified in order to cope with host mobility. Typically, the solution is based on host routes, with or without tunnelling. Tunnelling presents problems, e.g. its complicated interaction with some IP QoS protocols. **Path Updates**: This refers to the mechanism for installing information in the fixed network so that packets can be successfully forwarded to the MH at its new point of attachment. It can consist of the intelligent transmission of specific *update* messages or the use of modified Mobile IP registration messages.

Handover Management: This Issue looks at the impact of handovers on the MH (whereas the previous Issue took a network-centric view). Handovers should be fast and smooth, i.e. they should be performed without significant delays and without loss of packets. Also, soft handover may be allowed, i.e. a MH can simultaneously communicate with more than one BS at a time.

Support for Idle Mobile Hosts: Paging reduces the frequency of refreshments/updates for an idle MH in order to achieve two goals: reduce the protocol overhead (signalling, route lookups and memory requirements) in the network and minimise a MH's power consumption.

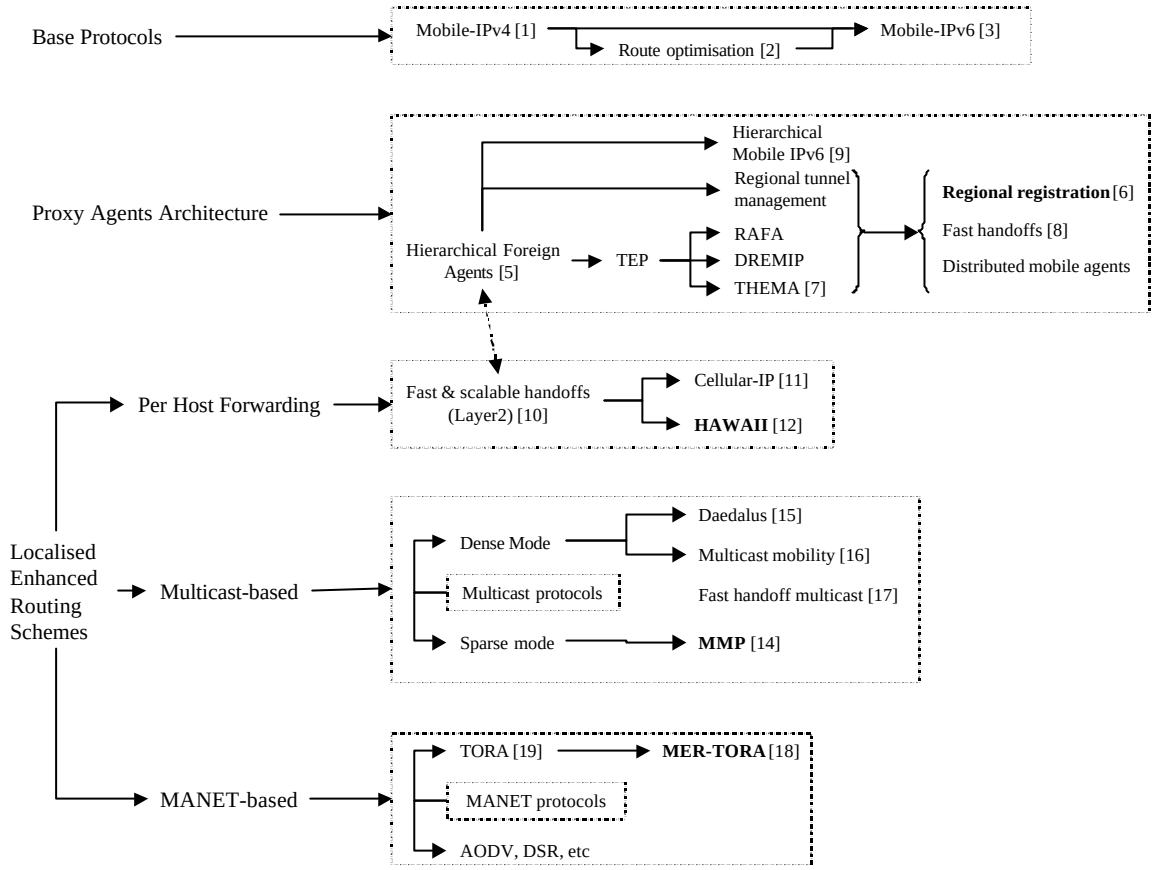


Figure A3-38: Classification of IP Mobility Proposals

Requirements for Mobile Hosts: An important decision is to what extent MHs are required to participate in the establishment and updating of the routing structure that enables mobility. A reference example can be Mobile IP where a MH is required to perform minimal operations: registering addresses, detecting movement and refreshing registrations.

Requirements for Core Network Interface: This issue defines the functionality in the gateway router of the access network. The Gateway is the transition point between the global and regional mobility and can include functions such as interworking between regional and global mobility, mapping of addresses, tunnel management, central control of mobility protocol mechanisms.

Address Management: A MH typically has to be provided with an IP address in a visited network. The way this is done can have an important impact on, for example, handover performance, scalability (because IPv4 addresses are a scarce resource), and deployability (private Home Addresses may need to be supported in corporate networks).

Routing Topology: This refers to a general static view of the access network nodes, whilst the other issues above more or less cover dynamic protocol operation. It refers to the arrangement of these nodes (e.g. whether they must form a tree hierarchy) and their required capabilities (e.g. whether they can act as normal IP routers and/or BSs). The routing topology has implications on the scalability and robustness of the system, e.g. robustness may be a problem if the access network hinges on a single gateway node. This Issue also relates to the reaction upon any failure of links or routers.

Security: Mobility, and wireless access in particular, introduce intricate security issues: the user's access to a visited network need to be authorised and the requests for path changes have to be authenticated; the user's privacy should be preserved; the access network's topology should be hidden from MHs; interworking of IPSec is required. The majority of IP-mobility schemes include security features or a framework for their realisation.

A3.7.4.2 Evaluation Criteria

In the second part of our Evaluation Framework we identify the evaluation criteria. Initially we have grouped them into 3 broad topics:

a) Efficiency

?? minimal packet delays and handover latency

?? no significant packet loss, reordering or duplication, e.g. during a handover

?? good throughput

?? optimised routing (including MH to MH case)

?? small signalling load over wired and wireless links

b) Scalability and robustness

?? support of a large number of fast moving MHs

?? support of a large number of serving nodes in a domain

?? support of a large amount of traffic per MH

?? resistance to extreme cases such as link or node failures, i.e. no single points of failures

?? resistance to errors e.g. over wireless links

?? resistance to routing loops and race conditions

c) Applicability/Ease of deployment

?? simplicity

?? compatibility with the standard Internet protocols

?? ability to support int-serv/diff-serv QoS protocols

?? ability to support dumb MHs that are Mobile IP compliant

?? ability to adapt to changes in the network topology

?? applicability of the same basic approach to both IPv4 and IPv6

A3.7.5 Initial comparison of IP-mobility proposals using our Evaluation Framework

We have made an initial application of our Evaluation Framework to compare the different classes of IP-mobility protocol described in Section 2. We decided it was easier to do this through a representative protocol from each of the different categories (Table A3-7), rather than to deal abstractly with the general characteristics of each category. The intention is to draw out the strengths and weaknesses of the various approaches, rather than to find the “best” in one particular class. A detailed description of how each of the selected protocols works can be found in the appropriate reference; in this paper we assume that the reader is reasonably familiar with them. Figure A3-39 outlines how MER-TORA operates.

Category	**Exemplar protocol**
Proxy Agents Architecture	Regional Registration [6]
Per Host Forwarding	HAWAII [12]
Multicast-based	MMP [14]
MANET-based	MER-TORA [18]

Table A3-7: Example Protocol for Each Category

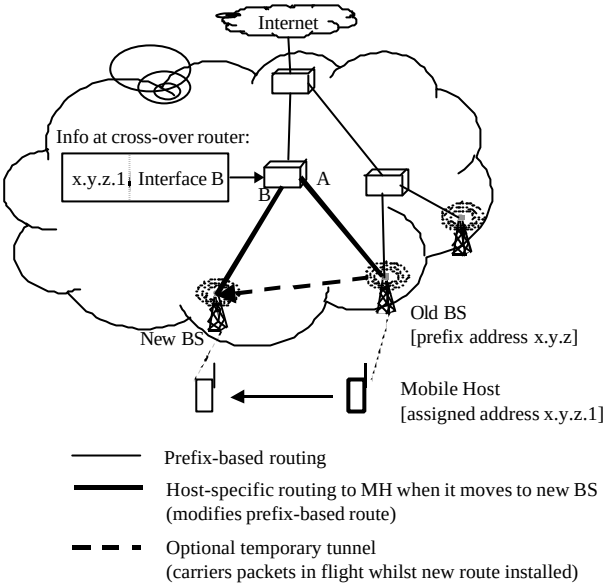


Figure A3-39: Mobile Enhanced Routed TORA

Table A3-8 summarises how our representative protocols tackle each Protocol Design Issue. This is followed by a discussion.

	Regional Registration	Multicast for Mobility Protocol	HAWAII	MER-TORA
Packet forwarding (downstream)	sequential tunnels	multicast forwarding (multicast encapsulation)	host routes for end-to-end encapsulated packets	prefix-based route to cross-over router; host-specific route below
Path updates	MIP + regional registration extensions (UDP)	CBT Join/Ack + ICMP (Instruct)	UDP Path Updates	UNICAST-UPDATE message from old-AR to new-AR for installing hard state, host-specific routes
Handover management	MIP, Route Optimisation	multicast join, advance registration, simultaneous bindings	Forwarding/Non-Forwarding schemes	localised at the edge of the network; inter-AR* tunnelling
Support for idle MHs	No	reduced signalling in wired network	paging using IP multicast	No
Requirements for MHs (in addition to basic MIP support)	I flag, registration keys as in MIP Route Optim., multiple level registrations	MIP Route Optim., multicast CoA	FA-NAI, MN-NAI, Challenge/Response, Route Optimisation	TORA, address acquisition, tunnel initiation, address return
Requirements for core network interface	HA must be able to handle the GFA IP Address extension	HA must accept registrations generated without an MN-HA authentication extension	HA must accept registrations generated without an MN-HA authentication extension	no distinction between 'global' and 'micro' mobility
Address management	Co-located CoA (bypasses the domain hierarchy), or FA-CoA	MH retains a multicast IP address within the domain. Ingress router seen as FA.	static Co-located CoA in foreign domain, Home Address in home domain	AR allocates an IP address from set it 'owns'. De-allocated at session end.
Routing topology	static configuration of enhanced MIP FAs in a tree structure	all nodes must support CBT IP multicast (sparse mode)	all nodes must be HAWAII-aware; standard routing protocols keep the default route up to date	all routers in a tree or in a mesh implement TORA (proactive prefix-routing + reactive host-routing)
Security	MIP + key distribution and authentication according to MIP-RO (FA-Key Reply extension) / DIAMETER	assumes Security Association between FA and HA	MIP + Challenge / Response or MIP-RO, password for path update messages, MN-FA and FA-HA authentication	use of existing mechanisms (RADIUS / shared keys / MIP+AAA)

(* The Access Router (AR) is the first IP-aware 'box'. For simplicity it is assumed this is the BS in the discussion below.)

Table A3-8: Summary of How Exemplar Protocols Tackle each Protocol Design Issue

We now discuss each Protocol Design Issue in turn, comparing our four exemplar protocols and drawing out points of interest. Our analysis is qualitative – thus we say only a little about the "efficiency" criteria, which is largely quantitative. We plan to remedy this later in the BRAIN project.

A3.7.5.1.1 Packet Forwarding

The main contrast here is between, on the one hand, Regional Registration and MMP which extensively use tunnels, and on the other hand HAWAII and MER-TORA which do not. Regional Registration forwards downstream data within the domain using sequential tunnels between FAs. This may be *inefficient*, although packet de-capsulation and encapsulation can be avoided by changing the IP addresses in the encapsulating header. With MMP packets are encapsulated by the ingress router into multicast packets and are forwarded using CBT interface-based routing. However, the major concern with tunnelling is that it obscures the original header, so making *applicability* of capabilities that depend on header fields more difficult (e.g. QoS). For Regional Registration, HAWAII and MMP, upstream packets can be forwarded with the same mechanisms that are defined for basic Mobile IP (e.g. using reverse tunnelling). On the other hand, MER-TORA uses the MER-TORA protocol for up and down-stream packets. In MMP packets destined for another MH within the domain are sent up to the ingress router, which reverses them back to the target MH.

A3.7.5.1.2 Path Updates

There are some interesting contrasts here. HAWAII and MMP both use soft-state path updates that are aggregated / merged as they travel up the tree, whilst MER-TORA uses hard state path updates[49]. Both methods aim to improve *scalability*. A quantitative comparison between them will be carried out later. Next, compare what happens as a MH changes its point of attachment: in MER-TORA it results in more host-specific state being installed (which 'over-rides' the prefix-based routes); whilst this is not so for the other schemes, essentially because their routing is entirely host-specific. Again, this will impact on the *scalability*, and the comparison may depend on how frequently the MH moves to another BS (for example). For both Regional Registration and HAWAII, a raceless (*robust*) and yet simple path management scheme is difficult to achieve if handoffs occur quickly [A3.82],[A3.101]. Because Regional Registration reuses the existing Mobile IP protocol messages, it can leverage on the recent enhancements to Mobile IP (e.g., for authenticating path updates), making its *deployment* easier. On the other hand, the scheme does not directly fit into the IPv6 mobility framework.

A3.7.5.1.3 Handover Management

All the protocols suggest conceptually very similar mechanisms for supporting fast and smooth handovers. Essentially, packets are forwarded from the old to the new base station after a handover and/or a route is set up to the new BS before the connection via the old one is lost. There is no obvious reason why one class of protocol should inherently perform better than another class. MMP has inherent support for simultaneous bindings through its advance registration feature, which may prevent packet loss during handovers; whilst HAWAII can optionally use dual-casting from the cross-over router, and it appears that this capability could also be added to MER-TORA if required. Regional Registration uses standard MIP move detection mechanisms, extended if necessary with fast handover support [A3.97] [A3.103] [A3.104], and smooth handovers as specified in MIP Route Optimisation [A3.78]. Similarly, both HAWAII and MER-TORA can optionally deliver, from the old to the new BS, packets that would otherwise be lost during handover. There are differences, however: in the Single Stream Forwarding sub-scheme HAWAII uses what it calls 'interface-based forwarding' which means that the outgoing interface (on which to forward the packet) is determined by both the IP address and the incoming interface, whilst MER-TORA uses a temporary tunnel. However, in MER-TORA if there is no tunnel when the link to the MH is lost (e.g. because handover is not predicted), then a virtual link is constructed to the MH from the old BS. It retains this for some time in the hope that it will be notified of the MH's new location. This virtual link should improve *robustness*, compared to the routing loops that can transiently appear in some HAWAII sub-schemes. There has been some work to try and quantify the *efficiency* of handover schemes, e.g. [A3.101] compared HAWAII to basic and route optimised MIP. However, there are no similar papers comparing all four of our protocol classes. We hope to address this within the BRAIN project.

A3.7.5.1.4 Support for Idle Mobile Hosts

Apart from HAWAII, paging seems to have received relatively little attention. Its proposal uses administratively scoped IP multicast [A3.89] to distribute paging requests to BSs. This should push paging to the edge of the access network, which assists in *scalability* and *robustness*. A similar scheme is probably widely applicable to other IP mobility protocols. MMP naturally tracks MHs as they move, through the standard messages to join to / prune from the multicast tree. It is suggested that the location management overhead may be able to be reduced for idle hosts by reducing the refresh frequency of the CBT "soft state" mechanism. A paging protocol has also been proposed for Regional Registration [A3.104]. The protocol aims at independence of link layer technologies; the MH agrees a 'sleep pattern' with the network, which requires synchronised sending of Paging Agent Advertisements from FAs belonging to the same Paging Area.

A3.7.5.1.5 Requirements for Mobile Hosts

HAWAII and MMP appear to have the *simplest* requirements on MHs, i.e. only MIP capability with extensions. However, a dumb MH might not be able to accept a multicast IP address as a CoA. In HAWAII the MH must be able to acquire a co-located CoA in a foreign network; in MER-TORA, suggests that a FA-CoA must be acquired. In Regional Registration the leaf FAs support basic MIP which guarantees the *compatibility* with dumb MHs.

[49] more accurately, hard state updates for the mobility related changes in topology, and both hard and soft state updates for non-mobility related changes.

A3.7.5.1.6 Requirements for Core Network Interface

The objective is to minimise changes to the standard IP protocols (e.g. at MIP HAs). All schemes seem to make some additional requirements on HA operation (limiting *applicability*); for instance, a HAWAII BS refreshes registrations with the HA on behalf of the MH, and these registrations do not contain a 'mobile home authentication extension', which might not be acceptable to a HA. MER-TORA can have several gateways (aiding *robustness* and *scalability*), whereas the others appear to be able only to have one. However, a deployment issue is that the backward *compatibility* of MER-TORA with MIP has only received limited consideration so far.

A3.7.5.1.7 Address Management

Address management is a key issue and a significant contrast between the protocols. With HAWAII, MMP and MER-TORA a MH keeps its IP address throughout the lifetime of the session (or longer), at least while it is in the same domain. This would (for example) ease the *applicability* of RSVP-based QoS support. By contrast, in Regional Registration the CoA changes at each handover. HAWAII requires that in a foreign network a MH acquires a publicly routable co-located CoA. Given the scarcity of public IPv4 addresses, this is a major drawback from the point of view of *scalability*. Also, because the CoA must be unique within a domain, a co-ordinated address allocation mechanism must be available. Regional Registration can also use a co-located CoA, and then similar comments would apply. But it can also use a FA-CoA and then IPv4 address exhaustion is not a problem. Within the domain, private CoAs can be used since they are not visible outside the domain. In MER-TORA, a MH is allocated an IP address by the BS (more accurately, the Access Router) where it starts a 'session', from the IP address block that the BS 'owns'. The pros are: fully prefix-based routing until the MH moves so minimising host-specific routing, and consistent address allocation across domain is *simple* since each AR owns its own address block. The cons are: more addresses are probably needed than for a IP mobility scheme with flat addressing across the domain, and more frequent address de-allocation is required (for *scalability* the IP address should be returned as soon as possible, e.g. at the end of an active session and not just when the MH powers down). If the number of MHs is large and their sessions short, then clearly a good, scalable DHCP implementation is needed. In MMP, the MH acquires a multicast CoA, so the shortage of IPv4 multicast addresses appears to be a major *deployment* problem. This should be less so in IPv6.

A3.7.5.1.8 Routing Topology

Clearly, the relevant routing protocol capability needs to be *deployed* in the nodes in the network. The effort is probably greatest for MER-TORA, because standard unicast routing (e.g. OSPF) is replaced by TORA. However, [A3.95] argues that it will give *scalability* advantages. *Robustness* is probably best for MER-TORA, since TORA was originally designed for mobile ad hoc networks (MANETs) so it will react immediately to any failure of links or routers. HAWAII relies on standard routing protocols for detecting failures; by integrating HAWAII with a routing daemon, a change in default route can trigger soft-state refreshes to HAWAII paths. Regional Registration and MMP would also rely on standard protocol recovery mechanisms to adopt to changes and failures. Regional Registration uses a central routing tree, whilst the others can have a tree or mesh topology.

A3.7.5.1.9 Security

Security has received limited consideration, especially for MMP and MER-TORA. In general, it is suggested that existing mechanisms can be used; for example, Regional Registration mostly refers to the existing Mobile IP related security infrastructure ([A3.98],[A3.99],[A3.100]). In MMP and HAWAII, the access network sends Registration Requests on behalf of the MH. These requests do not contain a Mobile-Home Authentication extension.

A3.7.6 Conclusions

In this paper we have proposed a Framework for the evaluation of IP mobility protocols, including the identification of Protocol Design Issues (which are the basic functional requirements) and the identification of Evaluation Criteria (against which the Issues can be assessed). We have suggested a classification scheme for IP mobility protocols, in order to recognise common characteristics of a particular Category and hence its strengths and weaknesses. Also, we assume it will allow a new protocol to be easily assigned to a Category.

We have presented an initial application of our Evaluation Framework. Rather than dealing abstractly with the general characteristics of each Category, we chose a representative protocol from each Category: Regional Registration, HAWAII, Mobile Multicast Protocol, and Mobile Enhanced Routing TORA. The results presented are only an initial examination using the Framework, due to early stage of our work. In

particular, quantitative criteria are mostly out of the scope of this study. For example, efficiency is difficult to evaluate because it should involve quantitative measures or simulation. We plan to deal with this later in our work.

From the discussion of the Protocol Design Issues it can be deduced that some bear more importance and complexity than others. Handover mechanisms and the interface between the mobile host and the access network entities appear surprisingly similar, whilst address management is a key differentiator.

Our goal is to use the Evaluation Framework to extract the best protocol mechanisms from all the investigated mobility protocols and to produce a clear perspective of the functionalities that need to be achieved by a new (or evolved) IP-mobility protocol, which we plan to propose at the final stage of our project. Another possible future direction could be designing a standard interface, or a standard architectural approach to IP micro-mobility. Already there is some effort in this direction: the Edge Mobility Architecture (EMA) [A3.95] and Open Base Station Architecture (OBAST) [A3.101], both of which aim to create a common approach to IP mobility whatever the wireless link technology.

A3.7.7 Acknowledgement

The authors would like to thank N. Asokan from Nokia Research Center and Prof. Hamid Aghvami from King's College for useful discussions.

This work has been performed in the framework of the IST project IST-1999-10050 BRAIN, which is partly funded by the European Union. The authors would like to acknowledge the contributions of their colleagues from Siemens AG, British Telecommunications PLC, Agora Systems S.A., Ericsson Radio Systems AB, France Télécom - CNET, INRIA, King's College London, Nokia Corporation, NTT DoCoMo, Sony International (Europe) GmbH, and T-Nova Deutsche Telekom Innovationsgesellschaft mbH.

A3.7.8 Paper References

[A3.77] C. Perkins, ed., "IP Mobility Support", RFC 2002, October '96.

[A3.78] C. Perkins, D. Johnson, "Route Optimization in Mobile IP", Internet Draft (work in progress), draft-ietf-mobileip-optim-08.txt, February '99.

[A3.79] D. Johnson, C. Perkins, "Mobility Support in IPv6", Internet Draft (work in progress), draft-ietf-mobileip-ipv6-12.txt, April '00.

[A3.80] E. Wedlund, H. Schulzrinne, "Mobility Support Using SIP", Proceedings of Second ACM International Workshop on Wireless Mobile Multimedia (WOWMOM), August '99.

[A3.81] C. Perkins, "Mobile IP", IEEE Communication Magazine, May '97 (contains Hierarchical Foreign Agents) or, C. Perkins, "Mobile-IP Local Registration with Hierarchical Foreign Agents", Internet Draft (work in progress), draft-perkins-mobileip-hierfa-00, February '99.

[A3.82] E. Gustafsson, A. Jonsson, and C. Perkins, "Mobile IP Regional Registration", Internet Draft (work in progress), draft-ietf-mobileip-reg-tunnel-02, March '00.

[A3.83] P. McCann. T. Hiller, J. Wang, A. Casati, C. Perkins, P. Calhoun, "Transparent Hierarchical Mobility Agents (THEMA)", Internet Draft (work in progress),draft-mccann-thema-00.txt, March '99.

[A3.84] K. El. Malki , N.A. Fikouras. S.R. Cvetkovic, "Fast Handoff Method for Real-Time Traffic over Scaleable Mobile IP Networks", Internet Draft (work in progress), draft-elmalki-mobileip-fast-handoffs-01.txt, June '99.

[A3.85] C. Castelluccia, "A Hierarchical Mobile IPv6 Proposal", Technical Report No 0226 INRIA, November '98.

[A3.86] R. Caceres and V. Padmanabhan, "Fast and Scalable Handoffs for Wireless Internetworks", Proceedings of ACM Mobicom, November '96.

[A3.87] A. G. Valko, "Cellular IP - A New Approach to Internet Host Mobility," ACM Computer Communication Review, January '99.

[A3.88] R. Ramjee, T. La Porta, S. Thuel and K. Varadhan, "IP micro-mobility support using HAWAII", Internet Draft, (work in progress), draft-ietf-mobileip-hawaii-00, June '99.

[A3.89] R. Ramjee, T. La Porta, and L. Li, "Paging support for IP mobility using HAWAII", Internet Draft (work in progress), draft-ietf-mobileip-paging-hawaii-00.txt, June ’99.

[A3.90] A. Mihailovic, M.Shabeer, A.H. Aghvami, “Multicast for Mobility Protocol (MMP) for emerging internet networks”, To appear in Proceedings of PIMRC2000, London, UK, September ‘00.

[A3.91] S. Seshan, H. Balakrishnan and R. H. Katz, “Handoffs in Cellular Wireless Networks: The Daedalus Implementation and Experience”, ACM/Baltzer Journal on Wireless Networks, ‘95.

[A3.92] J. Mysore and V. Bharghavan, “A New Multicasting-based Architecture for Internet Host Mobility”, Proceeding of ACM Mobicom, September ‘97.

[A3.93] C. Tan, S. Pink, and K. Lye, "A Fast Handoff Scheme for Wireless Networks", In Proceedings of the Second ACM International Workshop on Wireless Mobile Multimedia, ACM, August ’99.

[A3.94] A. O'Neill, G. Tsirtsis, and S. Corson, "Edge Mobility Architecture", Internet Draft (work in progress), draft-oneill-ema-01.txt,March ‘00.

[A3.95] V. Park and S. Corson, “Temporally-Ordered Routing Algorithm (TORA) Version 1 Functional Specification”, Internet Draft (work in progress) , draft-ietf-manet-tora-spec-02.txt, October ‘99.

[A3.96] P. Calhoun, H. Akhtar, E. Qaddoura, and N. Asokan, “Foreign Agent Keys Encoded as Opaque Tokens for use in Hand-off Process”, Internet Draft (work in progress), draft-calhoun-mobileip-fa-tokens-00.txt, March ’00.

[A3.97] P. Calhoun and C. Perkins, "DIAMETER Mobile IP Extensions", Internet Draft (work in progress), draft-calhoun-diameter-mobileip-01.txt, November ’98.

[A3.98] C. Perkins and P. Calhoun, "Mobile IP Challenge/Response Extensions", Internet Draft (work in progress), draft-ietf-mobileip-challenge-12.txt, June ’00.

[A3.99] C. Perkins and D. Johnson, “Registration Keys for Route Optimization”, Internet Draft (work in progress), draft-ietf-mobileip-regkey-01.txt, February ’00.

[A3.100] R. Ramjee et al., “HAWAII: A Domain-based Approach for Supporting Mobility in Wide-area Wireless networks”, www.bell-labs.com/user/ramjee/papers/hawaii.ps.gz, ‘99.

[A3.101] Discussion on OBAST in ‘cellular’ IETF mailing list. cellular@cdma-2000.org

[A3.102] J. Kempf and P. Calhoun, “Foreign Agent Assisted Hand-off”, Internet Draft (work in progress), draft-calhoun-mobileip-proactive-fa-01.txt, June ‘00.

[A3.103] K. El Malki and H. Soliman, “Hierarchical Mobile IPv4/v6 and Fast Handoffs”, Internet Draft (work in progress), draft-elmalki-soliman-hmipv4v6-00.txt, March ’00.

[A3.104] H. Haverinen and J. Malinen, “Mobile IP Regional Paging”, Internet Draft (work in progress), draft-haverinen-mobileip-reg-paging-00.txt, June ’00.

A4 Quality of Service Annex

A4.1 Introduction

The following annex provides a detailed overview of the QoS baseline architecture and the extensions proposed in section 4. The annex starts by outlining the interactions of QoS and the higher level protocols, mobility and the link layer, followed by an overview of the requirements and assumptions that were used when proposing the BRAIN QoS architecture. The evaluation criteria used to assess the architecture are then described, followed by a detailed description of each the extensions proposed to rectify weaknesses in the baseline architecture.

A4.2 Summary of QoS Interactions

QoS provisioning in the Internet involves protocol stack layers from the application right down to the link layer. Therefore, any network layer QoS architecture must consider the interactions with the layers both above and below it. The network environment through which QoS is being provisioned must also be taken into account. The following sections attempt to highlight these issues and interactions.

A4.2.1 QoS and Higher Level Protocols

The transport protocols provide some levels of end-to-end feedback concerning the QoS received by the application. This information can be used by applications to re-negotiate QoS within the network, and also by adaptive applications to modify their transmission rates etc. The following sections provide an overview of these mechanisms that are currently available in the Internet.

A4.2.1.1 Real Time Transport Protocol

The Real-time Transport Protocol was developed by the ``Audio-Video Transport Working Group'' and has recently become an Internet standard. RTP is described in the IETF's RFC 1889 specification as being a protocol providing end-to-end delivery services, such as payload type identification, time stamping and sequence numbering, for data with real-time characteristics, e.g. interactive audio and video. It can be used over unicast or multicast networks. RTP itself however, does not provide all of the functionality required for the transport of data and therefore applications usually run it ``on top'' of a transport protocol such as UDP.

RTP usually works in conjunction with a control protocol, the Real Time Control Protocol (RTCP), which provides minimal control over the delivery and quality of the data. RTCP provides support for real-time conferencing of groups of any size within an Internet. This support includes source identification and support for gateways like audio and video bridges as well as multicast-to-unicast translators. It offers quality-of-service feedback from receivers to the multicast group as well as support for the synchronization of different media streams. RTCP performs four main functions:

1. Feedback Information. This is used to check the quality of the data distribution. During an RTP session, RTCP control packets are periodically sent by each participant to all the other participants. These packets contain information such as the number of RTP packets sent, the number of packets lost etc., which the receiving application or any other third party program can use to monitor network problems. The application might then change the transmission rate of the RTP packets to help reduce any problems.

2. Transport-level identification. This is used to keep track of each of the participants in a session. It is also used to associate multiple data streams from a given participant in a set of related RTP sessions, e.g. the synchronization of audio and video.

3. Transmission Interval Control. This ensures that the control traffic will not overwhelm network resources. Control traffic is limited to at most 5% of the overall session traffic.

4. Minimal Session Control. This is an optional function which can be used to convey a minimal amount of information to all session participants, e.g. to display the name of a new user joining an informal session.

When an RTP session is initiated, an application defines one network address and two ports for RTP and RTCP. If there are several media formats such as video and audio, a separate RTP session with its own RTCP packets is required for each one. Other participants can then decide which particular session and hence medium they want to receive.

Overall RTP provides a way in which real-time information can be transmitted over existing transport and underlying network protocols. With the use of a control protocol, RTCP, it provides a minimal amount of

control over the delivery of the data. To ensure however, that the real-time data will be delivered on-time, if at all, RTP must be used in conjunction with other mechanisms and / or protocols that will provide a reliable service, RTP itself does not make any assumptions about the underlying network service. RTP does not address the issue of resource reservation or quality of service control; instead, it relies on resource reservation protocols such as RSVP.

A4.2.1.2 TCP and QoS

A congestion-managed rate-adaptive traffic flow (such as used by TCP) uses the feedback from the ACK packet stream to time subsequent data transmissions. The resultant traffic flow rate is an outcome of the service quality provided to both the forward data packets and the reverse ACK packets. If the ACK stream is treated by the network with a different service profile to the outgoing data packets, it remains an open question as to what extent will the data forwarding service be compromised in terms of achievable throughput. High rates of jitter on the ACK stream can cause ACK compression, that in turn will cause high burst rates on the subsequent data send. Such bursts will stress the service capacity of the network and will compromise TCP throughput rates.

One way to address this is to use some form of symmetric service, where the ACK packets are handled using the same service class as the forward data packets. If symmetric service profiles are important for TCP sessions, how can this be structured in a fashion that does not incorrectly account for service usage? In other words, how can both directions of a TCP flow be accurately accounted to one party?

Additionally, there is the interaction between the routing system and the two TCP data flows. The Internet routing architecture does not intrinsically preserve TCP flow symmetry, and the network path taken by the forward packets of a TCP session may not exactly correspond to the path used by the reverse packet flow.

TCP also exposes an additional performance constraint in the manner of the traffic conditioning elements in a QoS-enabled network. Traffic conditioners within QoS architectures are typically specified using a rate enforcement mechanism of token buckets. Token bucket traffic conditioners behave in a manner that is analogous to a First In First Out queue. Such traffic conditioning systems impose tail drop behaviour on TCP streams. This tail drop behaviour can produce TCP timeout retransmission, unduly penalizing the average TCP throughput rate to a level that may be well below the level specified by the token bucket traffic conditioner. Token buckets can be considered as TCP-hostile network elements.

The larger issue exposed in this consideration is that provision of some form of assured service to congestion-managed traffic flows requires traffic conditioning elements that operate using weighted RED-like control behaviours within the network, with less deterministic traffic patterns as an outcome. A requirement to manage TCP burst behaviour through token bucket control mechanisms is most appropriately managed in the sender's TCP stack.

A4.2.2 Possible Interaction between L2 and L3 QoS mechanisms

Schedulers are required at both L2 and L3 in order to provision the correct levels of QoS to applications. The number of queues available at the air interface is limited, while there can be many more at L3, e.g. one or more per application. Therefore, some scheduling is required at L3 to multiplex the data in the queues at L3 to the queues at L2, while ensuring the QoS for the data flows is met. Where a link-layer offers QoS support, it is desirable to take advantage of this. For example, if the link-layer can provision a number of logical channels, it is natural to map network layer flows onto these. These would clearly be expected to provide their own scheduling, and this would be used to maximise the delivered quality of service.

These schedulers must interact with each other in order to correctly control the amount of data passed to the link layer, e.g. in the form of queue size feedback. For example, if the queue at L2 is filling up because of a sudden drop in the availability of resources at the air interface, it is useful for this information to be fed back to L3, where the L3 scheduling can be modified to compensate and provide feedback to the appropriate application(s) provided.

The interaction between the schedulers at L2 and L3 is best controlled through the definition of service classes. This de-couples the implementations at each layer, which is important. The service classes provide a clean way for the network layer to request QoS from the link-layer, whilst keeping the network layer QoS transparent to the lower layers. Conversely it avoids the link-layer having to expose details of its own QoS implementation.

A4.2.3 Wireless Issues

Wireless networks have a number of significant differences to wired networks. Physically, wireless terminals have power restrictions as a result of battery operation. Wireless networks tend to have more jitter, more transmission delays, less bandwidth, and higher error rates compared to wired networks. These are all features that we would like to control within a QoS provisioned network. These features may change randomly, for example as a result of traffic or atmospheric disturbance. These features also change when a handover occurs. Additionally, may wireless networks are relatively expensive. A number of techniques have been developed to overcome some of these problems. However, in an IP network the interaction between the link layer and the higher layers should be minimised. Higher layers should not be communicating with the link layers, and protocols should not be designed to the requirements or capabilities of the link layer [A4.44]. Applications and transport layer protocols should not have "wireless aware" releases. Current implementations of wireless networks, including mobile phone networks and the wireless Internet (WAP) have all the above features [A4.45].

The large delays of wireless networks can lead to inefficient use of the network by higher layer protocols specifically with the transmission control protocol (TCP). The problems with TCP and wireless networks have been thoroughly studied [A4.46]. The focus of this study is on real-time applications that will typically use UDP transport.

A4.2.3.1 Memory in Mobile Terminals

There is no explicit control of jitter within the network, all jitter control is expected to be managed at the terminal through suitable use of buffers. This might be a problem as mobile terminals will have (relatively) restricted memory capabilities. However, simple analysis suggests that this is not a problem for the system described above. The maximum jitter comes from variations in the queuing delays. Thus the maximum jitter for the real-time service will be 120ms (See Section 4.) For a voice service at 9.6kbit/s this requires a buffer of 144 bytes. For a video service at 2Mbit/s, this requires a buffer of 30kbytes.

A4.2.3.2 Wireless efficiency

Wireless networks often have very restricted bandwidths, so there is a requirement to minimise the signalling overhead. However, as the project is focussed on the relatively cheap HIPERLAN/2 bandwidth, bandwidth optimisations are not considered as important as in traditional (regulated, restricted spectrum) mobile networks.

Wherever possible, hard state signalling protocols should be used. This minimises bandwidth requirements rather than processing requirements at the mobile and other nodes. Since network transmission is also a large drain on the mobile node battery, this is likely to be a good solution from the point of view of the mobile terminal.

Any reservation protocol, when used in hard state mode needs some modifications to ensure the safety of the network. One mechanism could be to use the data in a session to act as a refresh indicator for the session -an implicit signal that the reservation is still required. Additionally, nodes should monitor for ICMP "host unreachable" messages.

A further optimisation is to use one signalling message for several purposes - for example a suitably designed session initiation protocol could carry sufficient information to enable both the link and network layer QoS to be established. This type of protocol overloading and layer merging has been avoided in this solution. The reason for this is that HIPERLAN/2, and indeed other wireless LAN technologies, are not bandwidth limited unlike traditional mobile systems such as GSM or UMTS. It is further assumed that future wireless LAN technologies will be a significant mechanism for users accessing the Internet due to their use of unregulated spectrum. A further reason is that such a solution would lead to restrictions on how mobile terminals accessed the network, and would lead to complex processing within the network..

A4.2.3.3 Error Correction

Wireless networks have high losses. As well as random bit errors, they may suffer from complete packet losses – this is particularly likely during handover. This study assumes that preventing packet losses during handover is a responsibility of the handover mechanism. Wireless network manufacturers have developed mechanisms that provide error correction. Adding link layer error correction increases the delay experienced by traffic. The error rate on wireless links is so bad that it is a fair assumption that error correction techniques should be used wherever possible. As observed in [A4.46] this means that forward error correction techniques should always be used – this add redundancy to every transmitted frame in

order to enable data recovery and improve the bit error rate. Other techniques, such as automatic request repeat, can give much more accurate data transmission but only at the expense of much greater transmission delays. To enable any of these techniques to be used, network providers should assume that a significant proportion of any delay budget should be reserved for use within a wireless link.

Mechanisms also exist so that the wireless transmitters can control to some extent how the errors appear to the terminal. For example, some traffic (such as voice) prefers an even selection of bit errors to whole packet losses. If the link layer knows the type of traffic carried, it can control the loss environment by using different error correction schemes. In accordance with the principles outlined above, the solution proposed does not attempt to communicate wireless specific information from the application layer. However, the wireless base station could monitor traffic and provide optimisations as it sees fit. For example video is unlikely to be transmitted with less than 10kbit/s of bandwidth.

A4.2.3.4 Compressible Flows

The amount of bandwidth that a wireless layer needs to allocate to traffic can be drastically reduced if it can use header compression [A4.47]. This is particularly important for Voice over IP traffic using RTP/UDP/IP transport. There are three options to support this:

?? Pass application layer information to the link layer when making the QoS request

?? Allocate the full bandwidth request initially, but reduce this on detection of compressible (usually RTP) traffic. This may lead to reservations being refused unnecessarily.

?? Assume RTP will be used and under-allocate bandwidth for delay sensitive traffic. Since the vast majority of real-time traffic will use RTP, this may be a suitable solution -although the traffic will need to be monitored to detect and correct when this assumption fails.

A4.2.3.5 QoS link Layer Protocol

When QoS has been established at the network layer, once the traffic reaches the first router it is scheduled in order to achieve the required service. However in the wireless world huge problems could occur getting the data to the first router. Thus there needs to be a link layer mechanism that ensures that QoS is controlled across the first link into the Internet (an QoS MAC). This QoS protocol is link layer specific, and transparent to this study. One way to think about this link layer issue is to consider the base station and mobile terminals as elements within a distributed router. This then allows us to implement a link layer probe that is required by WP1 without breaking the transport network layered principles.

A4.2.4 Mobility Issues

The following section provides a discussion of the issues associated with provisioning and maintaining QoS in the mobile environment.

A4.2.4.1 Seamless Handover

Mobility procedures imply that the route taken by data will change. Any QoS that has been established for that data, and particularly any reservation, will therefore be disrupted. To ensure minimal disruption during handover, a number of alternative mechanisms could be used. These are discussed in order of increasing complexity.

For prioritisation QoS, little needs to be done to manage QoS during or after handover, as all class descriptions are relative.

The problem is more complex for reservation based QoS, where some service guarantees have been made. A reservation-based handover is described as seamless if the application or end user cannot identify that a mobility event has taken place. To some extent, this could be managed through careful descriptions of the service classes – for example by stating that traffic will be delivered within a certain time bound only 90% of the time! One improvement is that each node simply reserves a portion of its available bandwidth to be used solely for traffic that enters the node as a result of handover. This is known as a "static guard band". There needs to be a mechanism to enable nodes to identify the handover traffic and also the requirements of that traffic. One way to manage this in the ISSLL and DiffServ environments, where each packet in a QoS flow carries an explicit QoS class marking, is to assume that reservation-marked traffic entering a node in which it has no reservation is probably handover traffic.
This system can be improved upon when a centralised admission control system is used, as the size of this guard band can be adjusted dynamically. The nature of these policies, their complexity and the assumptions they make about user mobility are all areas under current research.

Without including specific QoS handover procedures, it is thus possible to engineer the network such that there is a high probability that, if the new route can support the handover, the handover itself will be seamless. However, it is also possible that 6 sessions simultaneously handover, of which 5 can be supported through reservation once handover is completed, but during handover all 6 suffer degradation. To avoid this situation, either the nodes involved in handover must communicate context information.

The alternative is for each session to make reservations for itself in nearby cells in preparation for likely handover. To ensure that this does not waste resources, these reservations are "passive" - the space can be used by best-effort traffic until the reservation is made "active". Since the mobile node should not know the network topology, pre-reservation is improved by making the base stations responsible for the passive reservations rather than the mobile node. Such pre-reservation schemes are difficult in the hop-by-hop admission control schemes, as the route through the network is not usually identified until handover has taken place. Therefore, such schemes couple the mobility management and QoS reservation process.

Specific implementations of these approaches are described in section A4.2.4.

Whilst temporary measures may suffice, there may be a requirement to repeat the call admission process, for example to establish a full reservation for traffic once route has stabilised. Ideally, this process should be confined to the region impacted by mobility. The RSVP local path repair process is an example of how this could be achieved. However, there the process has some weaknesses for the mobile and wireless environments, so a number of modifications are discussed in section A4.4.4.

A4.2.4.2 Micro Mobility mechanisms and IntServ

This section discusses the interaction between the micro-mobility mechanisms and IntServ. IntServ essentially implies the use of IntServ service descriptions and hop-by-hop call admission. Within the ISSLL framework, it is assumed that under-resourced regions of the network will be IntServ based. The areas where interaction is required between the micro mobility mechanisms and IntServ have been identified, and the following issues raised:

?? **Session creation**: when a new session is initiated, the resources must be reserved for the connection according to the requirements of the micro-mobility scheme. Associated issues include:

- How are the RSVP messages addressed to or by the mobile, tunnelled RSVP messages are not interpreted by intermediate routers
- How many reservations are required by the micro-mobility mechanism e.g. when a multicast based micro-mobility mechanism is used
- How are flows identified; RSVP cannot handle reservations for tunnels and the CoA for a mobile may change rapidly

?? **Session maintenance**: once the session has been created, any soft-state information in the routers must be maintained. The following issues need to be resolved:

- How often are refresh messages generated for both the RSVP reservations and any soft-state routing information, and how do they interact
- Can RSVP flows within the Access Network be aggregated to reduce signalling overhead
- How are errors handled within the Access Network

?? **Handover**: during handover, resources to the new base station must be available, preferably with the same QoS characteristics as the connection to the old base station. Associated issues are:

- Are the resources pre-allocated during session creation
- How is the old reservation released
- How far do reservation updates need to propagate back
- How can QoS be guaranteed after handover and does it interact with routing decisions

?? **Session termination**: when the connection is terminated the resources must be released, either explicitly or by waiting for them to timeout.

There are also issues concerning RSVP that may have an impact on the micro-mobility mechanisms, whilst having no direct interaction. These issues may be resolved by other tasks in Activity 2.2.

- How are duplex sessions handled

- Is there always an RSVP path to the host to handle signalling traffic, and how are non-RSVP enabled hosts handled at either end
- How are resources in the BAN managed if a new reservation to a base station is required
- Does QoS re-negotiation need to be supported, and how are changes signalled

Which scheduling mechanisms need to be supported within the network nodes to ensure QoS e.g. weighted fair queuing etc. Figure A4-1 shows a simplified BRAIN network containing a fixed host, A, and a mobile node, MN B. The grey boxes indicate areas where tunnelling maybe required.

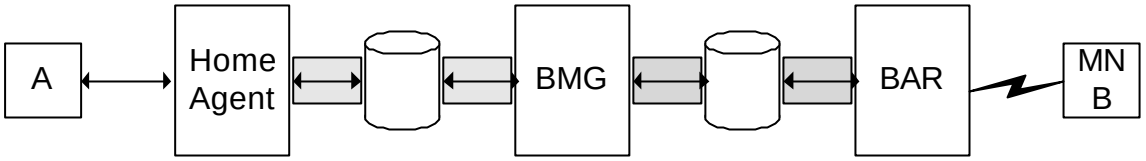


Figure A4-1: Overview of RSVP usage within BRAIN

IntServ may be used in the BRAIN architecture in any of the following ways:

1. End-to-end by applications to negotiate QoS across the network. In this scenario, A and MN B send RSVP signalling messages to each other using the home address of the mobile. The RSVP messages are transmitted transparently through the BAN and any tunnel between the Home Agent and the CoA node for the mobile. By itself, this approach guarantees no QoS treatment within the BAN because the RSVP messages are hidden within the tunnel.
2. End-to-end with mappings being carried out at the home agent, and possibly within the BAN, to allocate resources for the tunnel. This mapping of RSVP messages is not trivial because many of the objects within the RSVP message, such as the session and policy objects, will contain references to the home address.
3. Across the BAN only, so resources are allocated between the MN/BAR and the BMG in a manner transparent to the application on the mobile. In this scenario, the ingress network nodes must decide what QoS the data streams require by identifying the types of traffic in the streams and possibly using policy and user subscription information to allocate the appropriate QoS level. If an alternative signalling protocol is available and used by the MN to signal QoS requirements for the air interface, this information can be used by the BAR to determine the QoS requirements of the mobile. For reservations originating in an external network, the QoS might be indicated by an alternative mechanism, such as DiffServ, which can be mapped to RSVP reservations.

If optimal routing is used, the RSVP messages will not travel via the home agent, and will therefore not be tunnelled in the external network. Therefore, the reservation can be made end-to-end, as long as the RSVP messages are mapped appropriately by the end-points.

There are many architectural options for the use of IntServ in the BAN and one or more of these scenarios may be in use in the network. For the purposes of the following discussion, it is assumed that the events that trigger the creation of an RSVP reservation across the BAN are in place, and that messages that are intended for processing within the BAN routers have been mapped by an external entity.

IntServ can be thought of as consisting of two functions through which applications can choose among multiple, controlled levels of delivery service for data. The first function requires support from the individual network elements along the data path to control the QoS delivered to those packets, and is provided by the IntServ services such as Controlled Load and Guaranteed QoS. These are the QoS Control Services and describe how the network nodes along the route will treat each traffic flow. The second function provides a way to communicate the application's requirements and the QoS Control Service parameters to these network elements and can be referred to as QoS Signalling. In this discussion the QoS Signalling is provided by RSVP.

A4.2.4.2.1 QoS Control Services

The QoS Control Services relate to how different types of traffic should be treated by network devices. IntServ currently defines two types of service specifications: Controlled Load and Guaranteed QoS. RSVP messages transparently transmit information along the data path that is required by each network node to correctly invoke the QoS control services. The following section discusses the interaction of

micro-mobility mechanisms with the IntServ control services. The control service interaction issues discussed are common to all of the micro-mobility mechanisms.

Session Creation

When a reservation PATH message is sent from the sender to the receiver, each router along the path indicates the QoS control service that they can support, and to what extent. The receiver then uses this information to request a reservation that can be supported by the network that most closely matches the requirements of the application. The control information describing the QoS that can be supported by the reservation path includes:

?? a description of the traffic generated by the sender, the SENDER_TSPEC, which cannot be modified by intermediate routers.

?? data generated or modified by intermediate routers to indicate the services that are available along the data path from the sender to the receiver, and operating parameters used by specific QoS control services. This information is used by the receiver to make reservation decisions, and is carried in the ADSPEC object.

?? a description of the desired QoS control service, including parameters to define the traffic flow to which the reservation applies and how the service is invoked . This information is carried in the FLOWSPEC object, which is generated by the receiver and can be modified by intermediate routers.

This information is specific to the path via which the data will travel and will be routed through the BAN according to the micro-mobility protocol in use. For reservations to be successfully created across the BAN, hop-by-hop routing must be supported.

Session Maintenance

Once the QoS control service has been established, no session maintenance is required. When using RSVP as the signalling protocol, the reservations are soft state and need to be refreshed, but this is an issue associated with the QoS signalling protocol used by IntServ.

In the event of network node failure, the path from sender to receiver has to be modified, and there can be no guarantee that the QoS control service supported along the new route will be the same as the original. For example, the path may now pass through a router that does not support Guaranteed QoS, which will be indicated to the receiver by the information in the ADSPEC object. As a result, the reservation requested by the receiver will be modified to suit the new path, and the new reservation information will be propagated back towards the sender. Effectively, the reservation has been re-negotiated end-to-end.

Handover

On handover, the data path across the BAN is altered so that data is transmitted to the new base station. The QoS parameters can be installed along the new path in one of two ways.

1. The RSVP refresh messages will follow the new route across the BAN, and install the reservation along the new path. This method is simple, but does not guarantee that the QoS control service will be in place before data is transmitted along the new path, for example if a refresh message was generated just before handover and the time interval between refresh messages is quite long. This may adversely effect the data service that may already have suffered data loss during handover. The old reservation must also be removed and in some cases reservation overlap may occur.

2. If the QoS and micro-mobility mechanism is tightly integrated then the network entities within the BAN that are aware of the handover, and are able to route to both the old and the new base stations e.g. the crossover router in HAWAII, can set-up reservations to the new base station. The signalling required to set-up the new reservation can be contained within the BAN.

The problem of how to maintain the level of QoS is the same for both of the mechanisms above. The data is no longer traversing the same path and the services supported along the new route, either functionally or in terms of performance, may not be equivalent to the services along the old path. If the new path cannot support the same QoS as the previous route, then the QoS must be re-negotiated end-to-end, and the signalling cannot be restricted to the BAN.

A possible solution to the problem would be to have either QoS aware routing in the BAN so that bottlenecks can be avoided and routes that have plenty of free resources can be chosen. Also, if the mobile has a choice of base stations to which it can handover, the choice could be made according to which base station had the most resources available across the BAN. This information could be retrieved from a resource manager.

Session Termination

Session termination is handled by the QoS signalling protocol and the control service for the traffic flow is removed from the routers along the path.

A4.2.4.2.2 QoS Signalling

For IntServ QoS control services to be available along a data path there must be some mechanism by which the QoS information can be transmitted along the route. This discussion focuses on the use of RSVP as the signalling protocol used across the BAN and its interaction with micro-mobility mechanisms. In the following discussions, reservations set up from the external network or the BMG to the MN are referred to as downlink reservations, and reservations from the mobile to the BMG or the external network are referred to as uplink reservations. RSVP RESV messages will not reserve resources in a network node unless it is RSVP enabled and a PATH message has already been processed.

A4.2.4.2.2.1 Proxy Agent Architectures

Proxy Agent Architectures use a hierarchy of mobility agents to reduce the amount of signalling between the MN and the home agent. When the CoA of a MN is changed, the registration request does not need to travel all the way back to the home agent. The change in CoA need only propagate up the hierarchy as far as the change in route.

The mechanisms that will be considered in the following section are Regional Registration, Hierarchical Mobile IP (HMIP), and Fast Handoffs for MIPv4.

Regional Registration uses a proxy agent, the Gateway Foreign Agent (GFA), at the edge of a foreign domain, which is registered as the CoA of the MN. This CoA will not change when the mobile moves foreign agent under the GFA. There is a hierarchy of foreign agents beneath the GFA and the MN is allocated a CCoA. Each foreign agent in the hierarchy has a CoA and tunnels data to the next hop in the hierarchy determined by a mapping between the home address of the MN and the next CoA. This is shown in Figure A4-2.

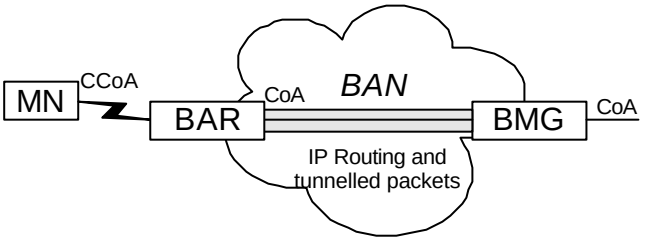


Figure A4-2: Regional Registration within the BAN

Hierarchical Mobile IP works in a similar manner to Regional Registration, but includes the concept of Private CoAs (PCoA) and Virtual CoAs (VCoA). The VCoAs are allocated to successive foreign agents within the hierarchy, and the PCoA identifies the link to which the mobile is attached. In Mobile IPv6, only the PCoA is required. This is shown in Figure A4-3.

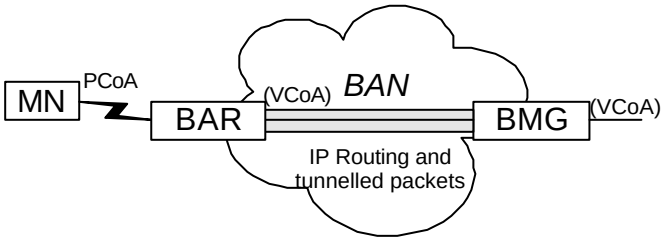


Figure A4-3: HMIP within the BAN

Fast Handoffs are a mechanism to support seamless handover when a MN moves between foreign agents. The traffic is bicasted to the previous base station and the new base station while the MN is moving between them. Simultaneous bindings are used to achieve bicasting of data.

Session Creation

For reservations in the downlink direction, the PATH messages will be generated or forwarded by the BMG towards the MN. At each mobility agent that the reservation passes through, the PATH message

must be altered so that resources are reserved for the next tunnel in the hierarchy. This process will occur iteratively down the hierarchy from the BMG to the BAR. The messages are routed using standard IP routing protocols and the CoA of the mobility agent at the destination end-point of the tunnel is used to identify the flow. The header of the tunnelled packets must also provide extra information so that the flow can be identified, e.g. include UDP header information so that the port numbers are visible [A4.4]. RESV messages generated in response to the PATH message will be routed hop-by-hop back towards the BMG, with mappings occurring in the mobility agents.

End-to-end uplink PATH messages are transmitted directly to the correspondent node. The source address is the home address of the mobile host in IPv4. In IPv6, the source address is the CoA of the MN, and the home address is indicated by the home address option in the packet. For reservations across the BAN only, which are invisible to the application, the BAR can generate the PATH messages but it must either know the IP address of the BMG that the traffic stream will travel via, or the BMG must intercept the RSVP messages so that the reservation remains local. RESV messages generated in response to the PATH will be routed hop-by-hop back towards the BAR.

For mobile-to-mobile reservations, where both mobile stations are within the same BAN, the reservation messages can be turned back by a mobility agent, if it has both MNs registered with it. Otherwise, the reservation will go via the destination home agent.

Session Maintenance

Refresh messages must be generated periodically to maintain the reservation across the network. The refresh messages will need to be mapped for each tunnel in the BAN. The frequency of the refresh messages is a configuration issue. ASSOC objects can be used to associate the refresh messages with the same session. Otherwise, reservation overlap may occur with multiple reservations for the same data flow.

The RSVP reservations between mobility agents can be aggregated to minimise the refresh messages required for maintenance. Aggregate reservations [A4.6] are implemented by sending the original RSVP messages transparently between the aggregation end-points and creating a separate reservation over which the aggregated data can be sent. This concept is similar to allocating resources for a tunnel and sending data transparently through it. Therefore, for each reservation between two mobility agents, the reservations can be aggregated to allocate resources for a tunnel over which all data passing from the source mobility agent to the destination mobility agent can pass. This concept is illustrated in Figure A4-4. In the first diagram, the two mobility agents have four tunnels between them, each with its own RSVP reservation. In the second diagram, the reservation has been merged to create one tunnel between the two agents.

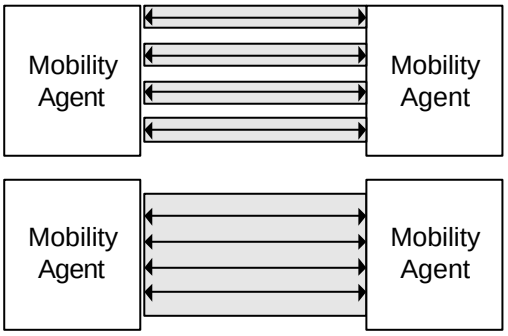


Figure A4-4: Aggregation of RSVP flows

To distinguish between QoS of the traffic flows within an aggregate reservation, it is suggested that DiffServ code points could be used to ensure that background traffic travelling through the RSVP tunnel does not interfere with real-time data. Alternatively, a set of tunnels could be used, each with a different QoS. This approach would be suitable for use with a QoS routing protocol where different types of traffic traverse different routes.

The routing protocols and the soft state nature of the reservations allow the BAN to recover from network node failure. However, the service might be disrupted while the recovery takes place.

Handover

When handover occurs, the CCoA/PCoA of the MN changes. The notification of the change in CoA will propagate up the hierarchy as far as the first mobility agent that already has a forwarding entry for the

mobile host. This node must be aware that it is the point where the reservations to the old and new base station for the MN converges, and must terminate the RSVP messages accordingly.

For downlink connections, a new reservation must be set-up as soon as the network becomes aware that the handover is about to occur. The network node that generates the PATH message will be the highest mobility agent in the hierarchy that is aware of the change in CoA. If fast handoffs are being used, then the reservation to the old base station must be left in place while the data is bicasted to the mobile. As soon as the old reservation is no longer required, it can be explicitly removed, or left to timeout.

In the uplink direction, the MN or BAR will set-up the new reservation as soon as either the mobile can signal its requirements, or the BAR knows that handover is going to occur. To minimise latency, it is preferable for the BAR to make reservations in advance of the handover, however the BAR must know or discover the reservations and QoS required by the applications on the mobile host.

Session Termination

The resources can either be explicitly removed or left to time out. The former option releases the network resources as soon as they are available and the admission control entity can be informed that the resources are available. The latter option reduces signalling in the network, but ties up resources for longer than necessary.

A4.2.4.2.2.2 Per Host Forwarding Schemes

The per-host forwarding schemes use path set-up protocols to install soft-state host specific forwarding information in the routers within the BAN. Each router has a default entry over which data can be forwarded to the BMG. The two protocols that will be considered in the following sections are Cellular IP and HAWAII.

Cellular IP maintains routing and paging caches in the routers via which packets can be routed across the access network. The caches map the MN's IP address to a route through the BAN in a hop-by-hop manner. The IP address of the BMG is the CoA of the MNs within the Cellular IP domain. The mobile host can only be associated with one BMG at any one time. Data and route-update packets maintain the entries in the caches.

HAWAII also maintains soft state, explicit routes across the BAN over which packets can be routed to the MN. HAWAII works in conjunction with Mobile IP, and the path set-up messages are triggered by Mobile IP registrations at a base station. Refresh messages are generated periodically to maintain the path information in the routers. It is recommended that the CoA is co-located to support QoS. Packets must therefore be tunnelled across the BAN. IP routing protocols are required in HAWAII-based BANs to build routing table entries, and to aid in recovery in the event of network node failure.

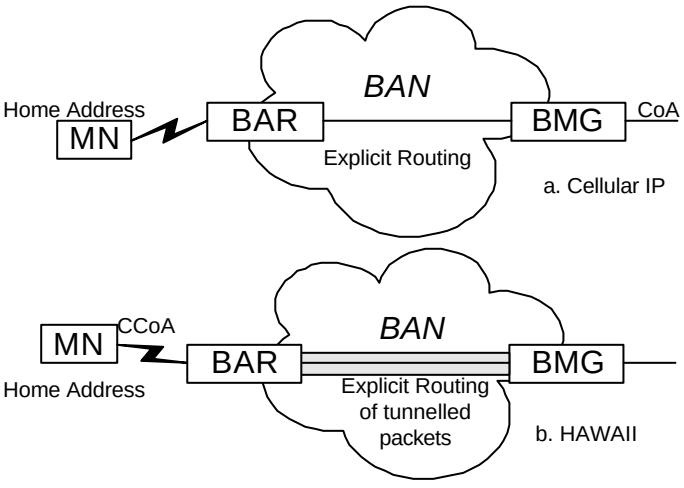


Figure A4-5: (a) Cellular IP within the BAN, (b): HAWAII within the BAN

Figure A4-5 summarises the use of Cellular IP and HAWAII within the BAN and shows the location of the CoAs.

Session Creation

In Cellular IP, the BMG is the terminating point for any tunnelled data from the external network for a MN. Therefore, PATH messages generated by an external node to initiate a reservation in the downlink

direction will be visible to the BAN routers as well. However, in HAWAII, downlink PATH messages must be addressed to the CCoA in order to reserve resources across the BAN.

Whenever a RSVP reservation is requested, the BMG or BAR must perform admission control on the request. If the network can support the reservation, the reservation is permitted. For reservations in the upstream direction, PATH messages are routed toward the BMG using the default route configured within the BAN routers. For end-to-end reservations in both mechanisms, the source IP address is set to the home address of the MN, and the corresponding RESV message is routed hop-by-hop back along the route followed by the PATH message. In Cellular IP, the routers within the BAN may not be addressed using the IP addressing scheme. If this is the case, RSVP will not function correctly unless hop-by-hop routing is supported and RSVP is able to associate previous and next hop routers with a traffic flow.

For mobile-to-mobile reservations where both mobile stations are in the same BAN, the PATH messages will propagate up to the BMG where they can be 'turned back' and treated as a downlink packet. If the BMG is not able to turn back packets, or the policy decisions prevent it, the reservation must go via the destination home agent, which would result in the reservation and use of resources in the external network when it is not required.

The RSVP reservation must follow the same route across the BAN as the explicit path set-up by the micro-mobility mechanism. One RSVP reservation is required per data session between the MN and the corresponding end-point.

In HAWAII, the CCoA of the mobile host remains constant while it stays within the domain so the RSVP flow can be identified using the CCoA plus other standard header information. In Cellular IP, the home address of the MN can be used.

Session Maintenance

For both Cellular IP and HAWAII, the forwarding entries are soft state and periodically need refreshing. Since the RSVP reservation is only valid while the forwarding path through the BAN is valid, the reservation should timeout when the route does. If the route is not maintained, the RSVP refresh messages cannot propagate across the BAN, and the reservation will timeout.

In Cellular IP, all data and route update packets that are sent by the MN refresh the forwarding entries within the routing cache. If the mobile has no data to transmit, it must send periodic route-update packets to maintain the cache entries. For end-to-end reservations, the mobile needs to generate periodic refresh messages to maintain the reservation across the network. In a tightly integrated solution, the RSVP messages can be sent with a frequency that will ensure that the routing cache entries do not timeout, so the route cache and the RSVP reservation can be refreshed by the same data packet. For reservations across the BAN only, where the reservation is transparent to the application in the MN, route-update packets from the mobile can be used to trigger RSVP refresh messages from the BAR towards the BMG. RSVP refresh messages must be generated periodically when the mobile is transmitting data as well. Note that RSVP sessions and route-update packets are only generated when there are active sessions on the MN.

In HAWAII, the soft state path information is maintained by path refresh messages generated by the MN. RSVP refresh messages should be sent at intervals no greater than those used to maintain the soft state route across the BAN, otherwise changes in the route will not be immediately reflected by the reservation.

The RSVP reservations can be aggregated from the base stations up to the BMG. A router within the BAN might detect that there are two reservations passing through it towards the BMG. Since both reservations will follow the same uplink route, it is possible to aggregate both reservations onto a single uplink reservation. This decreases the amount of signalling required in the network to maintain reservation by reducing the level of isolation between individual flows. Allocating DiffServ code points to different types of traffic could be used identify the QoS required by individual flows within an aggregate. However, the aggregation of RSVP flows, as proposed in [A4.6] requires the original RSVP messages to pass transparently through the aggregated path, and for a second RSVP reservation to be set-up to allocate resources for the aggregated traffic. This mechanism adds complexity to the routers within the BAN.

In the event of network node failure, HAWAII recovers using the IP routing protocols that will update the routing tables. The soft state nature of both RSVP and HAWAII ensures that path changes can occur in the BAN, but the time taken to re-route the data path may cause delays in the traffic stream and a reduction in QoS. In Cellular IP, network node failures destroy the explicit routes across the network, and there are no means by which data can be re-routed around the failed node.

Handover

On handover, a new reservation must be created to the new base station. The tightness of the integration of the per host forwarding scheme and RSVP has an impact on the amount of signalling and time required to set up the new reservations.

In Cellular IP, handover is triggered when a MN sends a route-update packet to the new base station. The route-update packet is forwarded to the BMG. For downlink reservations when the protocols are tightly integrated, the crossover router detects that the route towards the mobile has been changed and generates a PATH message accordingly. This will have the effect of gathering QoS information along the new route and triggering the generation of a corresponding a RESV from either the MN, for end-to-end reservations, or from the BAR for reservations across the BAN that are transparent to the applications on the MN. If there is no integration between the mobility protocol and RSVP, the resources along the new path will not be allocated until the sending node generates the PATH refresh message. This could introduce a high latency for the traffic if the reservation is not allocated immediately. In the uplink direction, the mobile host or BAR will generate a PATH message that will travel to the receiving node for loosely integrated or end-to-end reservations. Alternatively, the PATH message is intercepted by the BMG for reservations across the BAN only and possibly for tightly integrated solutions if the signalling is not required to refresh the reservation across the external network. The BMG will generate the corresponding RESV message to reserve the resources along the new route.

In HAWAII, if it is closely integrated with RSVP, the new reservation need only be made between the MN/BAR and the crossover router. For downlink reservations, the crossover router generates a PATH message along the new route. This will trigger the generation of a corresponding RESV message from the mobile or BAR. If the reservation was made in the uplink direction, the mobile or BAR will generate the PATH message that is intercepted by the crossover router, which generates a RESV in response. If there is only a loose integration between the protocols, then the reservation must be re-negotiated end-to-end. If the forwarding path set-up scheme is used, a temporary reservation must be created from the old base station to the crossover router. This temporary reservation lasts only as long as data is being forwarded from the old base station to the new base station and can be removed as soon as data from the external network is diverted to the new base station. The reservation must be made to ensure that forwarded data is received at the MN without significant additional delays, but has a high signalling overhead and requires extra complexity in the BAN routers.

Reservation overlap will not occur because the CoA of the MN does not change on handover.

If explicit tear down is used to remove the reservation to the old base station, then crossover router is responsible for issuing the required messages.

Session Termination

Once a session has been terminated, the resources allocated to the traffic flow can either be left to time out or explicitly removed. The former option means that network resources are allocated for longer than necessary and that the admission control entity must be informed by some network entity when the reservation has timed out.

When explicit call tear down is used, the gateway can inform the admission control entity when the call is released.

A4.2.4.2.2.3 Multicast-based Schemes

Multicast schemes use multicast addressing to forward data to one or more base stations to which the mobile may handover. They allow for location independent addressing and routing, and each mobile is allocated a multicast-CoA. The MN instructs neighbouring base stations to join or leave its multicast group, and data is forwarded to all of them. Only one base station forwards data to the mobile at any one time, while the rest buffer data for the MN. When handover occurs the old base station starts to buffer data while the new base station begins forwarding it to the mobile. This scheme aims to minimise latency and packet loss on handover for traffic sent in the downlink direction.

The Daedalus multicast proposal and MMP will be considered in the following sections. In the Daedalus scheme, the home agent is responsible for forwarding data to the multicast address, whereas MMP uses multicast purely within the BAN. In MMP, MIP is used for inter-domain mobility, and the home agent forwards data to a CoA, the address of the BMG, and the BMG forwards the data on to the multicast address that includes relevant base stations within the BAN.

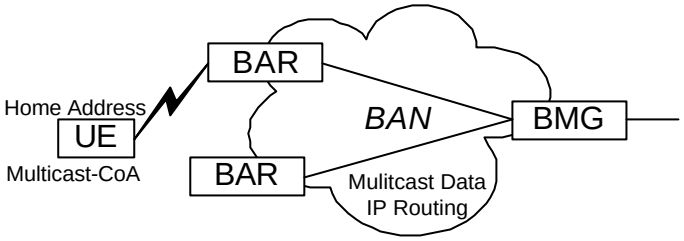


Figure A4-6: Multicast Scheme across the BAN

Figure A4-6 summarises how multicast schemes may work in the BAN. One BAR is forwarding data to the MN, while the other is buffering the most recent data sent to the mobile. These schemes require the support for multicast routing protocols within the BAN.

Session Creation

Reservations can either be made to all base stations within the multicast group, or just to the base station that is actively forwarding data to the mobile node. Admission control must be performed before reservations are permitted in the BAN. The first case creates reservations to multiple base stations, and allocates resources to base stations that are not actively passing data to the mobile. It may be the case that the mobile station never attaches to a particular BAR that is buffering data for it, rendering the resource allocations to this base station unnecessary. The second case eliminates the reservations to base stations that are buffering data but has the consequence that the data arriving at these base stations may suffer a greater latency than the data being forwarded to the mobile. Therefore, when the mobile changes base station the buffered data may be invalid.

In the scenario where only one reservation is created to the MN, the PATH message can be sent to the multicast group but only the base station that is actively forwarding data will send back a corresponding RESV. In the scenario where there is a reservation to every base station, each base station generates a RESV in response to the multicast PATH message.

The RSVP PATH messages may be forwarded onto the MN if RSVP is being used end-to-end. Otherwise, the base station can terminate them and an alternative mechanism to reserve resources over the air interface is required.

The traffic flow in the downlink direction can be classified on multiple fields using the multicast-CoA as the destination address and the address of the home agent or BMG as the source address.

End-to-end reservations in the uplink direction are unicast from the MN to the correspondent node and the source address is set to the home address of the mobile. For reservations across the BAN that are transparent to the applications on the MN, the BAR will send a PATH message that is intercepted by a BMG. The BMG will then generate a corresponding RESV message, which will be forwarded hop-by-hop back to the BAR.

Session Maintenance

The reservation must be periodically refreshed to prevent it from timing out. In the uplink direction, refreshes are periodically generated and transmitted in the same manner as for any unicast reservation. In the downlink direction, the PATH refresh messages will be sent to all base stations in the multicast group where either all base stations, or just the base station that is forwarding data to the MN, will generate a corresponding RESV. The frequency with which the PATH refresh messages are transmitted in the downlink direction can be integrated with the generation of the IGMP/MLD query messages.

When a new base station enters the multicast group, the IGMP/MLD messages can trigger the sending of a PATH message to ensure that the new base station can reserve resources if desired. In the case where all base stations set up reservation across the network, it is possible to merge the reservations to improve scalability. Merging occurs when the RESV messages for the same multicast flow meet within the same network node.

Network node failures are handled by the standard IP routing protocols and the soft-state nature of the RSVP reservations.

Handover

On handover, the base station to which the mobile is attached changes. In the uplink direction, this means that the reservation across the BAN will be routed via a different path. The reservation along the new

route will be created by the refresh messages generated by both end-points. The mobile could generate a PATH message as soon as it is attached to the new base station, or the base station could generate one as soon as it knows the mobile is going to hand-off to it. However, the base station needs to know the reservations required by the mobile. The reservation from the new base station may not be in place before the mobile wishes to transmit data.

In the downlink direction, if all base stations already have a reservation set-up, then the mobile can change base stations without having to create a new reservation and the QoS should remain consistent, as long as all base stations in the multicast group requested the same level of QoS. The reservation from the old base station must be removed, either explicitly, or by leaving the reservation to timeout. Otherwise, the base station must generate a RESV as soon as it becomes aware that the MN is going to hand-off to it. However, there are no guarantees that this reservation will provide the same level of QoS as the previous reservation.

Session Termination

The reservation can either be removed explicitly, freeing resources as soon as they are available, or left to timeout. Leaving the reservation to timeout reduces the signalling in the network, but allocates resources for longer than necessary. For explicit removal in the downlink direction, the teardown message can be multicasted to all members of the multicast group. If the reservation is end-to-end, the message is forwarded to the MN. In the uplink direction, the release of the reservation is the same for any unicast reservation.

A4.2.4.2.2.4 MANET-based Schemes

MANET-based schemes use routing algorithms that are able to adapt to changes is network topology as MNs move around the network. The protocol that will be considered in the following section is MER-TORA, and its suggested usage within the EMA.

MER-TORA creates and maintains a Directed Acyclic Graph rooted at the destination node. Each node within the graph is assigned a height and data can only travel 'downhill' towards the destination. When the MN attaches to a base station, it is allocated a CCoA, which it retains for the duration of the session while it remains in the same foreign domain. Each base station has a range of IP addresses that it can assign to MNs.

Session Creation

The RSVP PATH messages will follow the route through the network that was constructed by the MER-TORA protocol. For end-to-end reservations in the uplink direction, the mobile will generate the PATH message and send it directly to the correspondent node, setting the source address to its home address. For uplink reservations across the BAN, the BAR will generate the PATH message on behalf of the application on the mobile using information about the type of data to determine the required QoS.

For reservations in the downlink direction, the PATH message will follow the route to the MN/BAR, and the destination address must be set to the CoA of the MN.

The responding RESV messages are sent hop-by-hop back towards the source of the reservation. The asymmetric routing in MER-TORA is no different to the routing in standard IP networks, so the RESV packets must be addressed to the next hop node that has a corresponding PATH state installed on it. Therefore, while the packet may not travel directly to the next hop in the reverse direction, it will always arrive at the required next hop all the way back to the source, reserving resources along the uplink route. Intermediate hops that have not received the PATH message will ignore the RESV.

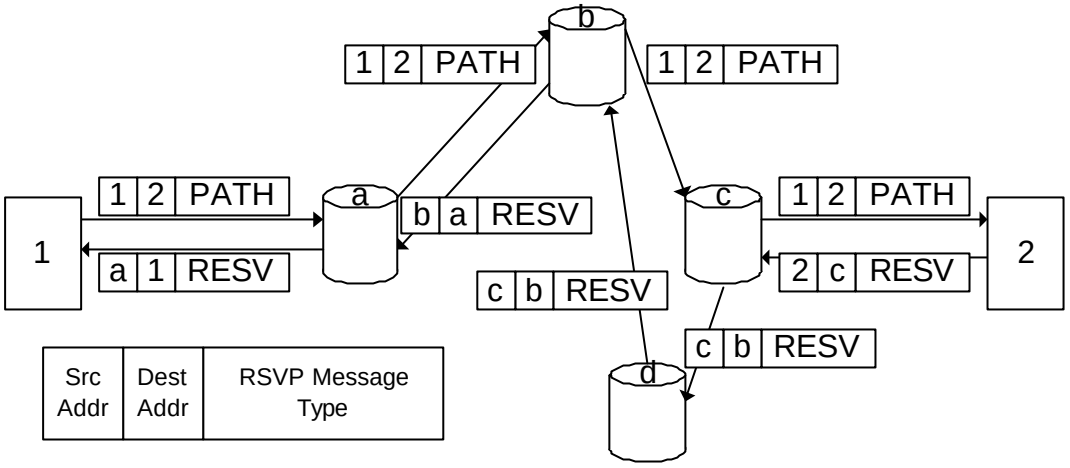


Figure A4-7: Simplified RSVP messages in a MER-TORA routed BAN

Figure A4-7 illustrates this concept. The PATH message is sent from node 1 to node 2. In the reverse direction, c knows that to send a packet to b, it must go via d. Therefore, the RESV will be routed by d to b, but d will ignore the RESV message because the original PATH message did not travel through it.

Session Maintenance

MER-TORA includes features that allow the routes to be optimised so that the shortest route to the network nodes are used. Two mechanisms that are used to optimise routes are:

1. Re-registration of the mobile: when the MN is idle, and attached to a BAR that did not allocate its CoA, the MN can hand back its original CoA and request a new one from the BAR to which it is currently connected. This will not effect RSVP reservations as it is only done when there are no active sessions on the mobile.
2. Use the optimisation (OPT) packet: this packet is periodically propagated outwards from the destination and, when received, resets the reference levels of all nodes to zero. Until the routing tables have been re-built, the reservations cannot be routed across the network. However, as soon as a route is available, the RSVP PATH and RESV messages will install the reservation state.

It is proposed in [A4.6] that the frequency of the RSVP refresh messages can be significantly reduced because of the hard state nature of the MER-TORA protocol. To achieve this, the MER-TORA and RSVP protocols must be tightly integrated. As long as the QoS along the path is not being altered, the refreshes within the BAN can be generated only when MER-TORA path update messages are received. However, if the QoS signalled by the PATH and RESV messages is different, e.g. after a handover, then the refresh messages must travel end-to-end in order to install the new QoS information across the network.

If a network node fails within the BAN, the MER-TORA protocol will re-build the routing tables according to the new network topology. The RSVP refresh messages will traverse the new route and install the reservations in each network node. The recovery from network node failure may disrupt the data service while the network nodes discover the new network topology. The reservation for the traffic flow, which may need to be modified from the original reservation, must be re-installed along the new path.

Handover

For downlink connections, when a mobile station moves base stations, a temporary tunnel is set-up between the old base station and the new base station over which traffic is forwarded until the routing tables have been updated with the new location of the MN. Unless the there are plenty of available resources in the BAN between the two base stations, the tunnel will need to have resources allocated to it, even though it is only temporary. This ensures that the traffic travelling between the two base stations meets the required QoS specification. As the routing tables are updated across the BAN the PATH messages, and their corresponding RESV messages, will traverse the new route and reserve the resources. However, these changes will take a finite time to take effect across the BAN and this could lead to a disruption in service. The new route will travel via the old base station, so none of the original reservations need to be removed. Only the resources allocated to the temporary tunnel need to be released once the routing table updates have occurred in the network.

In the uplink direction, the MN or the BAR, must generate the PATH as soon as possible in order to reserve the resources along the new route. If the BAR is reserving resources on behalf of the mobile in advance it must know the reservations and QoS required by the applications on the mobile. Before handing over, the MN or the old base station must explicitly remove the old reservation, or else leave it to timeout. If the reservation cannot be created in advance, there may be some disruption to the service while the new reservation is created.

Session Termination

Once a session has been terminated, the resources allocated to the traffic flow can either be left to time out or explicitly removed. The former option means that network resources are allocated for longer than necessary, and that the admission control entity must be informed by some network entity when the reservation has timed out.

When explicit call tear down is used, the gateway can inform the admission control entity when the call is released.

A4.2.4.2.3 Conclusions

?? **QoS Control Services**

The QoS control services relate to how different types of traffic are treated by network devices. These services were defined for fixed network environments, where routes across the network remain relatively stable. These control services do not include parameters to indicate the QoS required across the air interface, such as drop tolerance or bit error rate, and, therefore cannot take the quality of the air interface into account when setting up the reservation. One possible solution to this would be to define a new control service that would provide parameters about the quality of the air interface to the application. This would be carried opaquely in the RSVP message. Alternatively, RSVP could be deployed up to the base stations, and a proprietary protocol could be used to allocate the resources to the mobile host. This protocol, and a means of mapping from it to RSVP parameters, would need to be developed.

In the fixed network environment, once the control service for a traffic flow has been installed in the network nodes along the route, it is unlikely that the route will change for the duration of the session. This means that the QoS provided to the application will not be disrupted by a path re-establishment along a new route. In the mobile environment, however, handovers can occur frequently, which alters the route that the traffic traverses. This means that there is a period of time where the reservation for the traffic flow must be re-negotiated and re-established, during which traffic will receive a default best-effort QoS. This can lead to unacceptable delays for real-time traffic flows, with packets arriving late and out-of-sequence.

?? **QoS Signalling**

- Proxy Agent Architecture

  The proxy agent architecture uses tunnels between levels in the hierarchy. RSVP reservations must be mapped at each level to reserve the resources for each level. This leads to additional complexity within the network nodes in the BAN, and an increase in the signalling required to create the reservations. The tunnel reservations can be aggregated to reduce the signalling required, but this adds extra complexity to the routers and is not yet widely supported in commercial routers. Alterations to the reservation will disrupt the QoS for a traffic flow while the reservation is re-established. The severity of the disruption will depend on how many of the tunnel reservations need to be altered.

- Per Host Forwarding

  RSVP can be loosely or tightly integrated with the routing mechanisms of these protocols. If tight integration is used, the reservation can be updated at the same time as the route and the changes need only propagate as far as the crossover router unless the QoS is re-negotiated. Otherwise, the reservation will not be installed until a refresh message is generated. The tightly coupled approach ensures that the reservation is in place as soon as the route is installed, and minimises the disruption to the application data. In the loosely coupled approach, there will be a period of time during which there is no reservation along the new route, which means that the application data will receive a best effort service until the reservation has been installed. This will introduce unacceptable delays to real-time traffic.

- Multicast

The multicast mechanisms require network nodes to support the multicast routing protocols, IGMP or MLD. If reservations are set-up by each base station when it joins the multicast group, a QoS for the traffic flow can be guaranteed after handover, but at the cost of using more resources than are required. Otherwise, the base station can issue a RESV when it requires a reservation, but this may introduce a latency for the traffic until the reservation state is installed across the network. The RSVP PATH and RESV refresh messages can be integrated with the IGMP/MLD protocol query messages so valid members of the multicast group set up the reservations.

- MANET-based

  The MER-TORA and RSVP protocols can be tightly integrated so that the RSVP PATH and RESV refresh messages are only generated when there is a route change in the network. However, if the QoS needs to be re-negotiated for the new route, the reservation must be updated end-to-end. If the protocols are not tightly integrated, the reservation across the new route will only occur when refresh message is generated. This delay in setting up the reservation can impact the QoS for a session, because, until the reservation is in place, the traffic will receive only a best effort service. Another source of disruption to the service is the route optimisation feature of MER-TORA, which resets all the routing table entries and then re-builds them. The reservations will have to be re-negotiated and re-installed for every traffic flow, which will place additional load on the BAN and will delay the application data.

If RSVP is tightly integrated with the micro-mobility protocol, the QoS for the traffic flows is less disrupted than if they are only loosely integrated. To minimise the disruption further, it is desirable for the reservation to be set up in advance of handover. For this to occur, either the mobile must have some mechanism by which it can inform the new base station of its QoS requirements, or the base station must have a means to determine this information. To set-up or modify reservations with a minimum delay, the QoS signalling messages can be prioritised so that they traverse the network more quickly.

A4.3 Base Line Architecture

The following section provides an overview of the design choices made within the QoS group, and goes on to provide a detailed description of the chosen baseline architecture.

A4.3.1 Basic Design Choices

A4.3.1.1 End-to-end principle.

QoS is only useful, and therefore most likely to be paid for, if it exists on an end-to-end basis. This does not mean that the QoS mechanisms to provide a particular guarantee are the same across the network.

However in an increasing range of contexts, the end-to-end principle is being questioned. Certain large providers claim that they are able to charge their peer network providers for guarantees, without concern for the end customer leading to quality of service mechanisms that are being developed in ways that are inconsistent with end-to-end design. These changes are often justified as necessary for the supply of particular network services, such a virtual private networks. However, this is regarded as a short sighted which could restrict the future development of innovative services on the Internet.

A4.3.1.2 Classes of Service.

Different applications generate different types of traffic, and have different requirements on its handling in the network. There needs to be a way for the user to request service, and to understand what service is being offered. This is known as the class of services. Experience has proved that complete and flexible parameters sets, as provided by the Integrated Services architecture, are so complex that they are rejected by users. Additionally, complex classes may make network admission control and router scheduling management difficult, and sometimes inefficient. Alternatively, classes can be defined that are simpler to understand and met obvious application requirements.

Study of the wireless environment suggests that class definitions may not be the same in the fixed and wireless environments. For example, within the fixed environment loss and delay can both be controlled together, whereas there is a (non-deterministic) inverse relationship between delay and loss in the wireless environment.

QoS class definitions should define time periods for QoS measurements such that excess error messages are not triggered to the application. The definition of these classes is considered out of scope for BRAIN.

A4.3.1.3 Per Flow or Aggregation scheduling

Per-flow traffic management means that the application's traffic is granted resources completely independent of the effects of traffic from other traffic in the network. This enhances the quality of the service experienced by the application, but also imposes a burden on the network which needs to maintain state for each flow and to apply independent processing for each one. In the core of large networks, where it is possible to support millions of flows simultaneously, this traffic handling may not be practical.

When traffic is handled in aggregate the state maintenance and processing burden on devices in the core of a large network is reduced significantly. However, the quality of service is no longer independent of the effects of traffic. Allocating excess resources to the aggregate traffic class can offset this effect. However, this approach tends to reduce the efficiency with which network resources are used.

A4.3.1.4 Reservation vs. Prioritisation

QoS may be achieved through per flow reservation. Here an application queries the network to discover if the QoS requirements can be achieved. Reservations make best use of resources allowing a better planning of the network usage, and giving a more reliable QoS. However, there is a large overhead associated with this, with additional messages required and a delay before applications can start to send data packets.

The alternative, prioritisation model is where the client marks their packets to request a "premium" service when required. The user is able to make use of the service at any time, however the service provided may be less predictable and suffer when there is network congestion. Prioritisation is used with service definitions that may be general, or defined on a per-user basis.

A4.3.1.5 Signalling

For reservation based services signalling is required. The signalling may be carried with the data, in-band, or it may be separate from the data. In band signalling ensures that the information is always carried to

each router that the data visits, which is useful when routes change frequently as in mobile networks. However, it also means that an overhead has to be carried in every data packet. For voice traffic in particularly this overhead is large approximately 10% of the packet size.

The signalling may be soft state, which makes it resilient to node failures, or it may be hard state, which can minimise the amount of signalling. Where hard state signalling is used, a different set of mechanisms must be introduced to cope with node failure and protect the network

A4.3.1.6 Bi-Directional Reservations

Different models exist about responsibility for generation of the signalling messages. These models are often coupled with responsibility for payment for use of the network. In one model, the mobile node is responsible for establishing the required Quality of Service through the mobile network domain for both outbound and inbound traffic. This model does not require that both ends of the communication share the same understanding of QoS signalling. It is a useful solution to providing QoS in a bottleneck wireless network region. However, it is less easy to provide true end-to-end QoS in this situation. It is difficult to provide such a solution when inbound and outbound data paths are asymmetric. Other solutions have one party responsible for establishing the QoS over the entire end-to-end path. The standard Internet models assume that the receiver is usually responsible for QoS establishment, as they receive value from receiving the data. However, these solutions usually require that the data sender also participate in any signalling and they retain ultimate responsibility for any payment – this is seen as a possible mechanism for limiting "junk mail".

A4.3.1.7 Traffic Classification and Conditioning

Once data is transmitted there are a number of functions that need to be provided to ensure that the network is protected against malicious use. As in call admission, these functions may be provided on a hop-by-hop basis, or solely on entry and exit to a network. By using these functions on exit from a network (and terminal) we can ensure that transmitted data is within the contract, so that the behaviour through the network is understood.

Classification identifies the flow to which traffic belongs, through analysis of the packet header. The packet can then be associated with a particular QoS contract. Once the flow has been identified, meters measure its temporal properties against the QoS contract. One action that may be triggered by the measurement is traffic shaping. This is the process of delaying packets within a traffic stream to cause it to conform to the traffic profile. A packet marker might be used to label traffic to ensure it receives the required QoS treatment through that network domain. Additionally, packet markers might change the marking on a packet if the traffic has broken the agreed contract. This re-marking also acts as a signal to the receiver that the QoS contract was violated – enabling action to be taken by the end-to-end application. Packet droppers, which simply drop packets, provide another means to handle traffic that has broken the agreed contract.

The basic architecture does admission control at the edge routers only. This can only be used to give a limited level of QoS guarantees, particularly in an access network where, there will be low levels of statistical multiplexing. It does not allow any mobility related information, such as handover probabilities, to be taken into account. This can lead to weak service guarantees and inefficient networks. Two solutions are here proposed.

A4.3.1.8 Hop-by-hop Admission Control for traffic aggregates

Here, each router is responsible for its own admission control decision. This hop-by-hop admission control can lead to very strong service guarantees, and more efficient network use. Coupled with the Internet resilient routing, this enables a system that has no single point of failure. However, this solution also leads also to more state held within the network and more processing is required within the routers, so this solution does not generally scale well. However, for certain types of services, specifically those based around bounded delay concept, many of these problems can be avoided. Annex reference. This service, which still routes on traffic aggregates, has been studied to define the regions in which the per-flow admission gives significant efficiency improvements, and it can easily revert to the edge based admission as the system becomes larger. One outstanding problem with hop-by-hop solution is that since no node has global knowledge, they cannot take account of the load of nearby routers, from whom they may receive handover traffic

A4.3.1.9 Centralised Admission Control

Here, edge routers refer all QoS requests to a centralised admission control unit, which instructs the routers how to behave. This scheme, which enables more accurate resource handling and flexibility in resource assignment, also enables basic QoS management even when the core routers have no QoS

support. It allows the call admission mechanism to be upgraded and replaced easily. Certain types of call admission criteria, in particular delay based admission, are less well suited to centralised admission schemes. This is because the centralised unit can never know the actual full state of the network [A4.18]. Similarly, centralised admission schemes may be less suited to the mobile environment where the state of the network is likely to change rapidly, although such a system can take account of the state near-by nodes when calculating admission control decisions. Finally, any solution based on a centralised node, needs careful design to prevent scalability and reliability problems.

See also section A4.4.2, for an in depth study of Bandwidth Broker Approaches to centralised call admission.

A4.3.2 Description of Base Line Architecture in depth

The proposed architecture is based on the work done in the IETF ISSLL working group and specifically on the proposal "Framework for Integrated Services operation over DiffServ Network" [A4.20] an architecture for allocating resources in a Differentiated Services network using the RSVP protocol. In addition, the issues raised by the Internet Architecture Board [A4.24], have been taken as basis for this proposal.

The proposed architecture has been created to answer the need for an IP-based scalable QoS architecture that allows flexible support for mobility of terminals in a cellular network. The fundamental design criteria have been to use the existing IETF protocols and architectures where possible and to add new extensions if needed. The mobility of terminals and the QoS signalling and management are viewed as closely related but separate tasks, thus both have their own architecture.

The DiffServ architecture is simple and scalable in the sense that only aggregate traffic is checked at boundary of networks and no per-flow information state and processing is needed in network core. However, fairness is hard to maintain within aggregation, therefore the nature of the service offered by DiffServ is more approximate in nature than the service received through RSVP reservations.

A4.3.2.1 Background for the proposal

The building of the QoS architecture should start from the top of the protocol stack, from the application requirements. WP1 has defined two types of QoS support: explicitly signalled QoS requirements and more direct, not beforehand signalled, transfer with QoS support. The former turns into an application level protocol and the later turns into a pure transport layer implementation. The most widely used application level QoS signalling protocol is at the moment the RSVP protocol, which will also be the basis for the "*signalled*" transfer of data. In view of architecture and protocol deployment, it would be most beneficial, if the QoS signalling protocol is widely known – trying to establish a new QoS signalling protocol as an end-to-end signalling protocol would be close to an impossible task, at least in a short period of time. The "*non-signalled*" transfer of data using the transport layer implementation could well be a system based purely on the Differentiated Services framework and the use of DiffServ Code Points (DSCP). As it happens to be, the IETF has defined an architecture and methods for using RSVP to signal QoS request into a DiffServ network, thus allowing at some level the concurrent use of both of these protocols [A4.20].

The IETF has a wealth of protocols and architectures for delivering and co-ordinating QoS-related services - It would be of little use to start defining new protocols and totally neglect the previous work. Therefore, the primary aim of this proposal is to use the existing techniques and, if needed, add some modifications to them. A further goal has been to take into account the issues raised by the Internet Architecture Board [A4.24] regarding the present QoS architectures and their weaknesses.

A4.3.2.2 Protocols and architectures.

The BRAIN QoS architecture is based on the ISSLL framework. We use the Integrated Services Framework (IntServ) [A4.23] and the Resource Reservation Protocol (RSVP) [A4.4],[A4.25] as the signalling method for explicit resource reservations. RSVP reservations are mapped to Differentiated Services (DiffServ) [A4.22] forwarding classes at the edge of the network and forwarded according to standard DiffServ operation [A4.21]. To RSVP, the core access network between the BAR and the BMG is a single link.

In addition to the per-application signalled reservation, the architecture also allows flexible DiffServ Code Point (DSCP) marking for applications that are not able to quantify their resource needs, but would benefit from better than best-effort service for the data exchanges, RTP-based VoIP flows, for example, or HTTP traffic.

There are basically three resources to be shared in a BRAIN network: the radio resources in each cell, seen by the BAR's IP layer as a network interface; the access network resources between the edges of the network; and the resources on the interface between the BMG and the external network.

From another point of view, the RSVP-based flows get a circuit-switched like connection, while the DiffServ marked flows get a priority-enhanced packet-switched service. In a circuit switched connection, there is always a connection set-up phase prior to the data transfer. When RSVP is used, the default virtual circuit switched connection is unidirectional and, therefore, two set-up phases are required for a guaranteed bi-directional connection. The DiffServ marked transfer is less reliable due to the lack of setting up a dedicated reservation but faster to initiate and does not require support from the correspondent node, as with RSVP.

An important decision in the architecture is to leave the uplink flow marking to the MN. This has two main advantages. First it takes part of the packet handling away from the BAR; the BAR does not need to check for the IP and transport headers for multifield flow classification and can do faster Behaviour Aggregate classification. Second, it allows the MN to use IPSec payload encryption (and any IP-within-IP tunnelling). If the original IP packet was encapsulated into another IP header, the BAR might not have enough information to do multifield classification. The MN is free to choose the proper marking within the code points provided by the network operator. The standard DiffServ code points must however be available, but operator specific code points can be used.

The downlink flow marking is based on SLA-driven information related to different application flows, e.g. HTTP, FTP, and RTP. The gateway has SLA information for incoming flows, which can be compared to the incoming packets to find the right DSCP values.

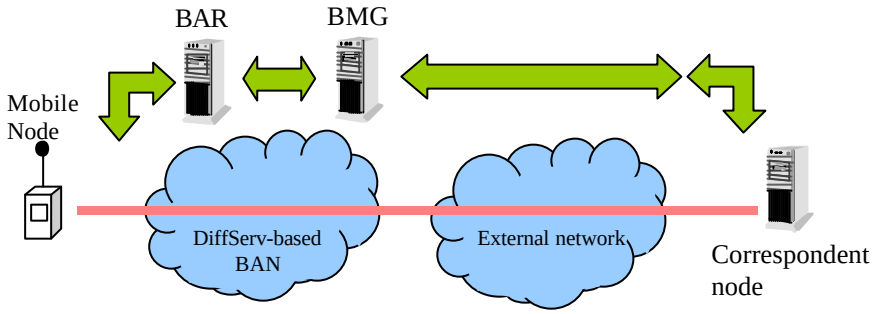


Figure A4-8: QoS signalling in the Network Nodes

A4.3.2.3 Network Nodes

The network nodes in this architecture and the signalling flows are presented in Figure A4-8. The BRAIN Access Router (BAR) node is the first (last) IP-based node to which a flow originated from (terminating to) a mobile node arrives, the closest IP-based node to the MN. The BAR is in charge of resource co-ordination for the access points[50] under it.

The BAR node has much of the same functionality as a DiffServ edge node upgraded with functionality needed to support RSVP signalling and mapping the signalling to proper DiffServ Per-Hop Behaviour (PHB) aggregates. The service is based on the SLAs negotiated between the subscriber and the ISP and can be varying according to the present time and date and network load. The BAR must provide mapping tables derived from the SLAs. Dynamic negotiation of SLA's as defined by e.g. [A4.19] seems to be a useful future enhancement.

The BRAIN Mobility Gateway (BMG) has much the same functionality as the BAR but for flows arriving from the external networks. Admission control functionality controls the traffic arriving from the external ISPs, dropping or remarking packets if they do not conform to the SLA for the mobile being reached.

The access network internal routers forward packets according to normal IP routing mechanisms and the DiffServ processing [A4.21]. No assumptions are made how a BAN operator provides the requested services. If the BAR and BMG perform proper shaping of flows admitted into the network and the core access network is over-provisioned it may even be possible to operate BAN internal routers without any active QoS differentiation features.

[50] By Access Point we denote a layer 2 only device through which IP packets are forwarded transparently.

The benefits of the proposed architecture are that it allows both signalled (RSVP) and non-signalled (DiffServ) QoS-aware transfers using standard protocols. This allows for a wide range of mobile terminals to communicate with an even larger range of correspondent nodes. By using DiffServ in the intermediate routers the routing and QoS control create minimal overhead. The architecture is not however particularly optimised to support mobility, apart from having aggregate packet forwarding that provides flexibility in allocating resources on changing paths. Other schemes presented later will enhance this basic architecture to better support mobility and give a more options for QoS resource handling and signalling.

A4.3.2.4 Session management

In order to support different kinds of applications and usage scenarios, the network must be flexible in its understanding and allocation of resources to applications. We can identify the following kinds of data transfer scenarios:

1. MN-originated transfers (both MN-to-MN and MN-to-CN)
   - ?? When the correspondent node is RSVP aware
   - ?? When the correspondent node is not RSVP aware
2. CN-originated transfers
   - ?? When the correspondent node is RSVP aware
   - ?? When the correspondent node is not RSVP aware

Note that this presentation discusses a CN external to the BAN. When two BRAIN terminals request resources between each other, the session management operation is similar. If the terminals are within the same BAN, the signalling will not go out of the BAN.

A4.3.2.4.1 MN-originated transfers

In the default case, since we are discussing QoS-aware networks and applications, the CN is QoS and RSVP aware. When the MN initiates the request for a certain QoS, the MN sends the RSVP PATH message. This message arrives at the BAR, which stores information about the request and forwards the message further on to the BMG. The BMG stores a similar state and forwards the message to the external network.

Once the resulting RESV message arrives from the CN to the BMG, it checks for resource availability. If resources are available, the BMG will forward the message to the BAR, otherwise the BMG will cancel the reservation according to standard RSVP processing. When the BAR receives the RESV message, it will perform similar operations and if resources are available, it can record the RSVP-to-DSCP mapping to be used if the MN is not marking the upcoming flow. The mapping information can be available at the BAR for direct use, or the BAR can request the SLA for the MN from an external entity like a Bandwidth Broker.

If the CN is not RSVP-aware, or the MN's application itself is not RSVP-aware, the user could still be able to request a certain service for the applications transfers. The user could be able to request some better than best-effort service on a per-application basis. The protocol stack could then set some well-known DSCP to the packets that originate from the given application and thus trigger a better forwarding behaviour from the ISP. The MN-based DSCP marking could also be based on the receiver/sender port numbers and addresses, for example, a connection initiation to the well-known port 80 could be an indication of a WWW-request, and the user could set a low-to-medium priority (and cost) to those flows. A connection to the port 20/21, an FTP-transfer, could by default get a best-effort service, for example, and a Telnet connection could get quite a high priority due to its interactive nature but small amount of data transferred.

The DSCP marking can be done on both the MN and the BAR. If the MN does not do the marking by itself, the BAR can provide some flow distinction based on the negotiated SLA to this mobile.

A4.3.2.4.2 CN-originated transfers

A downstream reservation is similar to the MN-originated upstream resource reservation. The CN sends a PATH message that will go through the BMG and BAR and store a reservation state in these nodes. The MN studies the arrived reservation request, calculates the needed resources and responds with a RESV message. If resources are available, the RESV message will eventually reach the CN and the resources

within the BAN and possibly network between the have been reserved. In downstream reservations, the BMG needs to do the flow marking.

If previously unknown packets arrive from the outer network, the BMG again needs to do admission control against the SLA of the MN being addressed. The DSCP mapping can be by default to the best-effort service class, or the SLA can indicate some other mapping for different well-known service, like ftp, http, or some VoIP call. This is how non-signalled flows can be forwarded with some higher level of service. This requires the BMG to do multi-field classification, which requires more processing power on the BMG.

A4.3.2.5 Signalling plane

The following figure 5 summarises the signalling performed in the architecture. The MN interacts both through the application data flow and the RSVP-protocol with the CN. The BAR and BMG nodes also interpret the RSVP messages. The application can also use direct DSCPs to mark its traffic. This is considered to trigger services only from within the BRAIN network. The DiffServ markings may or may not persist end-to-end, therefore the flow is marked with a dashed arrow. The BAR and BMG can also signal with an optional Bandwidth Broker about SLA management

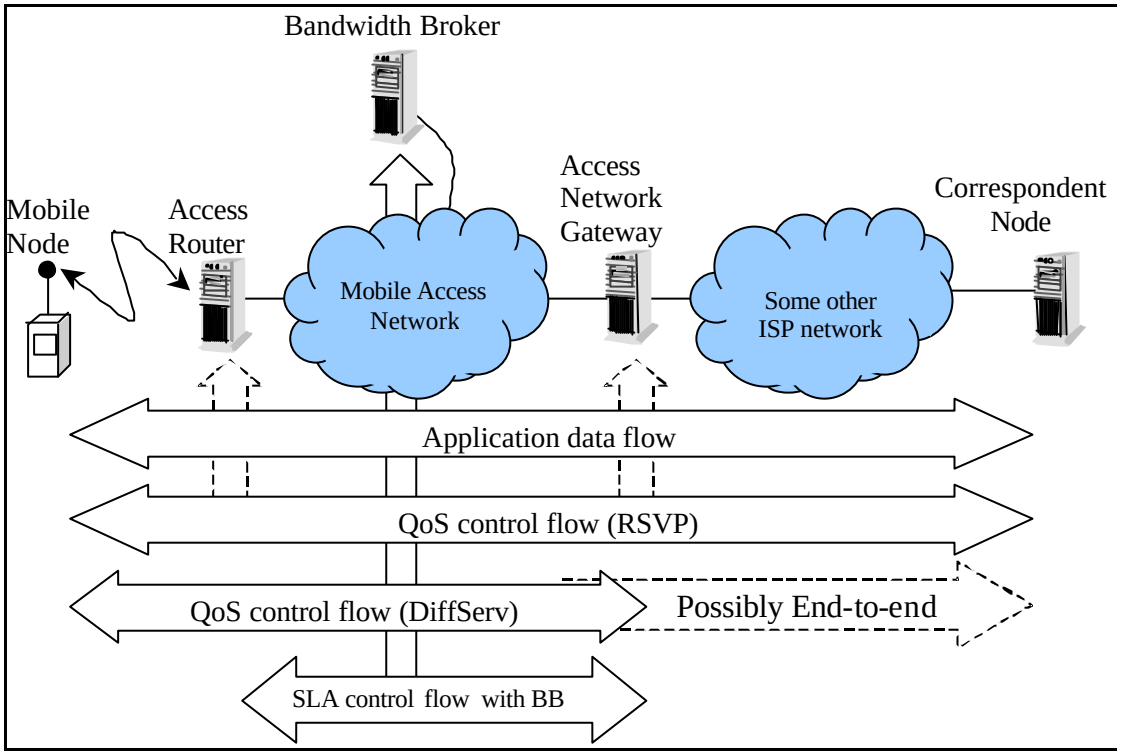


Figure A4-9: QoS Signalling

A4.3.2.6 Service set

The RSVP protocol does not fix the actual QoS parameters and the type of QoS the application can request. Applications can have different types of needs, which it can request service for. The requested service profile must then be mapped to DiffServ behaviour aggregates, to DSCPs and the resulting PHB. However, to make the mapping simple, at first, we can identify only two or three parameters that could be supported: defined measurable bandwidth and a delay/priority-based forwarding. The simplest service an application can request would be a given bandwidth. Network nodes can be aware of their available bandwidths from which it would be easy to calculate the remaining capacity allowed to new users. The "awareness" can be based on periodical calculations of the load or on static SLAs. The BB can be made aware of the resources at each node using the COPS protocol, and this information can then be propagated to edge nodes in order to allow for new flows and do a proper mapping to DSCP values.

An application could also request some assured delay or priority to its flow. The issue of assured delay can be divided into two guarantees: per-hop and end-to-end assurances. The per-hop assurance can be provided with the present DiffServ PHB models. Providing an assured end-to-end delay requires knowledge of the route of the flow and the performance of each router in the path. It would be possible to still provide those end-to-end guarantees, but that would require each router to reserve most of the

resources for this type of service. This would result in very poor total utilisation of router resources, because effective delay guarantees could only be given to small fraction of traffic while still reserving many times more resources in order to really guarantee the service. The issue of providing a PDB with delay guarantees has been studied in [A4.26].

Another way of providing a strict end-to-end delay guarantee would require forwarding the handling time from router to router. This type of information forwarding would require a new protocol and new headers, not to mention the performance implications, thus in order to make things simple, we leave the option out.

Instead, it would be possible to use the EF PHB aggregate that would include higher priority at each router (as in [A4.26]), thus higher priority traffic would be forwarded first, resulting in lower total delay.

Along with the explicitly signalled request for resources, a MN would also need, for flexibility and faster operation, to mark by itself packets with a certain DSCP in order to trigger some well-known service. Services that would benefit of such a direct marking include emergency services (in some countries this is a strict requirement) and SMS-type short message services. Also, if the application is not aware of any QoS, the MN stack should be able to trigger best-effort service automatically, either by setting a proper DSCP (or by just leaving the value to "zero"). Similarly, a DSCP resulting in highest possible priority (EF PHB), and cost, could also have a well-known DSCP.

As for example, typical web browsing using HTTP [A4.27] creates a large number of small TCP connections. If the application would need to request through (or the lower layer protocol stack would perform on behalf of the application) explicit QoS signalling some desired service from the network, the performance of web browsing would be seriously affected.

The BAR and BMG need to keep state of the number and size of flows running through using certain DSCPs. This is needed in order to do admission control also for directly MN-marked flows, which did not use RSVP. It would not be a good idea to allow more flows into the network, than the network can support.

The exact mapping of RSVP-reservations to DiffServ per-hop behaviours is not defined in this work. The mapping is very much related to individual access networks, and an operator may want to use specific internal code points to define non-standard per-hop behaviours within its own network. Those proprietary code points should however be mapped to standard values at the edge of the network in order to allow for better interoperability. Moreover, the resource allocation between service classes is also out of scope of this work. However, the following DSCPs (in no order of priority) could provide useful:

1. Best-effort
2. SMS-type short transfers
3. Highest priority, premium service
4. RTP
5. Network signalling
6. RSVP-support DSCP

In addition, it might be feasible to define a code point that would support mobile nodes that change their point of attachment to the network. A suggestion would be to reserve some fixed amount of resources for moving MNs. The resources could be taken into use with a dedicated DSCP for incoming users. When a mobile enters a new cell, it would use the resources put aside for some short period of time. During that time the MN could signal its actual need for resources, and then release the temporary resources for the following incoming users. If resources would not be available, the mobile would soon be signalled about the situation. This issue is left for further study.

A4.3.2.7 Mobility management

So far we have not discussed thoroughly mobility issues. Mobility of terminals is a two-level issue: Mobile IP-based macro mobility and micro mobility within an access network. The primary issue that arises with QoS flows and mobility of terminals is preserving the service requested and given to the user and his applications. In this proposal, the state of the network, the resource availability, is stored in two or three nodes: the BAR and the BMG, and possibly the BB. Mobility of terminals will result in flows to and from the mobile to change BARs and BMGs, which will require new resource signalling.

The core QoS architecture supports mobility as far as the standard IP protocols can support mobility by default. RSVP refresh messages are used to update reservations on new paths, but only between a lengthy time interval. Not much can be done to enhance DiffServ-based QoS guarantees in a mobile environment without specific mobility enhancements, since DiffServ does not provide explicit guarantees and does not have a signalling mechanism.

A4.3.3 Error Reporting

All the mechanisms discussed within the BRAIN project have been standard techniques. The interested reader is therefore referred directly to the following documents as starting points for further research.

[A4.4] for information of RSVP error reporting

[A4.22] for information on DS error handling

[A4.28] for information on ECN

[A4.48] for information on RED.

A4.3.4 Evaluation

A4.3.4.1 **Assumptions, Requirements and Limitations**

The following appendix provides an overview of the assumptions, requirements and limitations for the proposed QoS architecture and the extensions.

1. The BAN must provide QoS for both inbound and outbound traffic without requiring changes to other networks and terminals. However, the type of QoS that is achievable without currently available, standards based co-operation from other networks and terminals will be fundamentally limited, and may be further limited by the BAN implementation.
2. If BRAIN-specific QoS mechanisms are supported, their scope must be restricted to the BAN.
3. Resources should be protected against malicious use as far as possible. Typical problems could be related to theft of service and denial of service
4. The session/application layer signalling is transparent to the BAN, so the BAN QoS mechanism can not assume the presence of any specific type of session layer signalling and application layer QoS messages are not read by the network. However, there are no restrictions on session layer network devices such as RTP translators and mixers.
5. Original transport QoS parameters should be carried end-to-end regardless of the transport QoS protocol within each network. End-to-end RSVP signalling must not be broken if external networks and terminals support it. If RSVP information is available, it should be exploited. Additionally, we should try to avoid having terminals with 2 RSVP implementations within the terminals.
6. The BAN must provide QoS for generic IP hosts, i.e. hosts that are BRENTA unaware. It can be assumed that support for the IP2W functionality will be present in the mobile terminal.
7. Support for QoS re-negotiation must be provided.
8. It must be possible to interwork with non-QoS capable external networks and correspondent terminals. This follows from the fact that the BRAIN network is a standard IP network, so the basic best effort service is always guaranteed. However, a further requirement for some types of operator is that it is possible to provide both reservation and reservationless QoS within the BAN, for both in and outbound traffic, under control of the mobile terminal, even if both the adjacent network and correspondent terminals are QoS unaware.
9. If the external network or and correspondent terminals is QoS capable, the end-to-end QoS mechanism should operate and interact correctly with BRAIN mechanisms. This is limited to IntServ-style RSVP or DS networks.
10. There is an issue that exists about exchanging radio-specific information between network layer elements. Certain requirements may only be partially met if such information is not exchanged, however this leads to a risk of requiring “BRAIN specific” applications. Ideally, all radio specific information should be restricted to the sub-IP layers, or should be accessible through static configuration (such as the familiar preferences). As a minimum, radio and mobility enhanced QoS signalling protocols that may be supported by the BAN are restricted to the BAN. Similarly mobility

and radio enhanced QoS parameters that provide additional may be supported in, and are restricted to the BAN.

11. A mechanism by which the mobile terminal can signal its QoS requirements to the BAN must be provided. The basic interfaces are not BRAIN specific, thus any IP terminal with standard Internet QoS software will be able to access QoS within the BAN. BRAIN specific enhancements may be offered to enable increased functionality or better performance. A new end-to-end QoS protocol should not be used, unless there is a clear route to standardisation of this protocol.
12. The implementation of QoS in the BAN should not be visible to the mobile terminal. The interface shall only reflect the "bearer requirements" that a MN can request. Any mechanism or combination of mechanisms can be used together to provide the overall QoS in the BAN.
13. The IP2W layer should be invisible to the application. The behaviour of the system can not be guaranteed should the application interact directly with the IP2W interfaces.
14. QoS violations detected at the network layer must be reported asynchronously to the application layer in the terminal that requested the QoS behaviour. A QoS violation must be reported if QoS is not achievable on an end-to-end basis, even if QoS is successfully established within the BAN.
15. Fragmentation and re-assembly are likely to exist at the link layer
16. Seamless hand-over should be supported, once resource allocation has been assured. The aim is that, once a reservation has been given, the application is unable to detect any QoS changes that could be a result of the handover process. This is in part related to the classes definition – as an extreme, delay and loss could be time averaged such that a QoS violation does not occur even if all signal is lost during handover! Another aspect of this is traffic engineering, to ensure that sufficient capacity is available for handover traffic. These two issues have not been considered explicitly within this study. This study has concentrated only on the mechanisms to achieve seamless handover assuming the network has been well dimensioned and that this dimensioning is in part reflected in the service class definitions.
17. QoS mechanisms within the BAN may have enhanced support for mobility. Also, mobility mechanisms may have enhanced support for QoS. In particular, it is assumed that tunnels nodes are responsible for QoS within the tunnel
18. "context information" may be available to the handover mechanism to facilitate a context aware handover. This context information may include QoS state.
19. It is assumed that service level agreements exist
20. Currently, it is not clear which QoS signalling mechanism is preferred by the market. Therefore, the determination of a single BRAIN QoS signalling mechanism is out of scope. A couple of alternatives will however be discussed and evaluated.
21. Expecting no global QoS signalling mechanism soon leads to anther problem: a roaming BRAIN MN may initially not know how to signal for QoS resources while attached to a visited BAN. The mechanisms to do this are out of the scope of this discussion. The QoS signalling mechanisms described later may provide ideas how to solve the problem without being required to describe details. Ideas may be a directory polling mechanism allowing the MN to learn which signalling mechanisms are supported and preferred by the BAN. It may then be an option to download software required to signal QoS requirements. Another idea may be to map all currently available IP QoS mechanisms to the link layer and provide no network layer QoS signalling at all.
22. Mechanisms and an architecture to measure the QoS a flow receives are out of scope.
23. The link layer must identify and adjust to changes of link layer characteristics not resulting in QoS reservation changes, without requiring network layer signalling (also during hand-over).

A4.3.4.2 Evaluation Criteria

A list of criteria is defined that can be used to compare different architectural concepts for a BRAIN Access Network. The idea is that we in the end shall be able to find the best concept from aspects related to scale-ability, performance etc. For many of the criteria it will not be possible to put measurable figures against them (like maximum delay). In stead we need to describe outstanding feature and limitation of a given concept.

The criteria list may never be complete. It will be an iteration process to improve the list.

The focus will be on performance for the BRAIN domain related to:

?? Micro mobility (mobility within a BRAIN domain)

?? Delay related parameters for real time conversational services

?? Other QoS related parameters like packet loss etc.

?? Resource management including “resource protection” to prevent vicious attacks.

?? Resource related parameters (that can be mapped to meet the radio interface requirements)

Observe that the evaluation criteria cover the whole architectural domain of BRAIN and the criteria are not only focusing on performance but also to verify that we cover the right functionality level.

I have tried to segment the criteria in relation to different terminal states and the transitions between them. The following states are considered.

?? Disconnected

?? Passive

?? Active

The states and the possible transitions may be described as follows

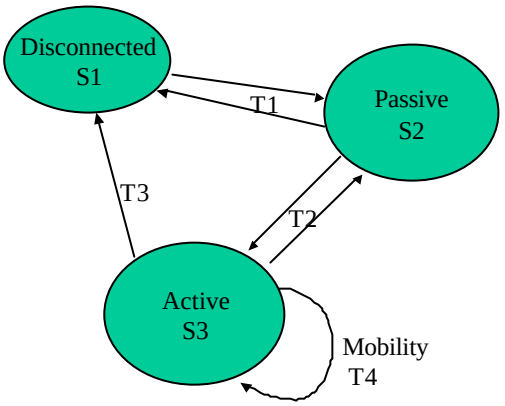


Figure A4-10: Mobile Terminal States

The transitions between Disconnected and Passive and between Passive and Active may all be controlled transitions. The transition from Active to Disconnected may only be accidental. No transition is assumed directly from Disconnected to Active.

Definition of states:

S1. Disconnected

The terminal is completely invisible to the BRAIN network. The WLAN port in the terminal is passive. The terminal can not be alerted over the WLAN interface.

S2. Passive

The network knows (believes) that the terminal is out there. The terminal can be alerted. Certain signals are accepted over the radio interface without pre-reservation or pre-activation. Hand over is not an issue.

S3. Active

Resources are reserved over the radio interface between terminal and network. The network knows to which base station the terminal is connected. Here we have different levels of resource reservation. From a lose reservation for a best effort service to a more strict reservation for a conversational service. The hand over mechanisms may be different for the different communication levels (at leased the requirements are different).

The transitions to be evaluated are:

T1. Disconnected to Passive (and the reverse)

The terminal identifies itself as available in the network. The home and visiting location registers are updated.

T2. Passive to active (and the reverse)

The terminal is connected to specific base station. The terminal is given an IP address. An anchor point may be defined (we may have different anchor points for different QoS classes). Resources are to be allocated.

T3. Accidental disconnect

Resources is to be released and location registers is to be updated even thou no contact with the terminal exist.

T4. Mobility

The terminal moves from one Base station to another. Locally a new path and associated resources with correct QoS has to be established and the old path has to be released.

A4.3.4.2.1 State criteria

A4.3.4.2.1.1 A1 Disconnected

The terminal is not reachable.

?? No requirements exist

A4.3.4.2.1.2 A2 Passive

The network knows (believes) that the terminal is out there. The terminal can be alerted.

?? What's signalling load over wired and wireless links

?? What's the power saving mechanisms (are there any?).

A4.3.4.2.1.3 A3 Active

Resources are reserved over the radio interface between terminal and network.

?? What's the signalling load (to maintain resources and QoS)? Shall be minimised.

?? To minimise the power consumption is maybe only a radio interface issue in this state?

?? What's the delay for user data from host to host and host to Gateway (distant dependent delay may be excluded)

?? < 10 msec for a conversational voice service. (The delay that can be allocated to a BRAIN Access Network, BAN, have to be a fraction of the service requirement for host to host communication. The Del 1.1 service requirement is 200 msec including non BRAIN network and distant dependent delay)

?? Error rate (this is maybe only radio interface related)

?? Bandwidth may have to be renegotiated (have to be supported). What's the complexity and load for such action?

?? Change of QoS class may be considered (may require re-establishment and re-routing of a reserved path).

?? What's the functionality for protecting a path for vicious attacks? Where in the BAN do we perform necessary filtering (at which hierarchical level?).

A4.3.4.2.2 Transition criteria

A4.3.4.2.2.1 General

The main issues under transition criteria will be to answer questions like

?? The time it takes to make the transaction. (Much of the transaction may not be noticeable by the user.)

?? The amount of signalling needed

?? Is there a potential for improvement? What can be suggested? How realistic is the improvement?

?? What is local?

?? What is assumed to be end-to-end?

?? What's the BRAIN access signalling when a BRAIN terminal wants to communicate with a non-BRAIN terminal (network) using a different access "protocol"?

A4.3.4.2.2.2 T1 Disconnected to Passive (and the reverse)

Location registers have to be updated locally and at "home".

?? What's the delay before the host can access another host or Internet?

?? What's the delay before someone outside can alert the host?

?? What's the load on the network for entering "passive state" and for going back to "disconnected state"?

A4.3.4.2.2.3 T2 Passive to active (and the reverse)

The terminal is connected to specific base station. The terminal is given an IP address. An anchor point may be defined.

?? What's the delay before the terminal can access the Internet using Best Effort?

?? < 0.5 second (delay in this order will considered as immediate reaction and will avoid the user to repeat his command. The Del 1.1 requirement is 3 sec including Internet authentication.)

?? Are (can) resources to be allocated over the radio interface for a best effort service?

?? What's the delay before you have established path with a QoS required for a conversational service?

?? < 0.5 second (delay in this order will considered as immediate reaction and will avoid the user to repeat his command. The Del 1.1 requirement is 10 sec Host to Host including session level signalling.)

?? What's the load on the network for doing this?

A4.3.4.2.2.4 T3 Accidental disconnect

The host is lost.

?? How long time does it take for the network to detect the lost terminal?

?? What's the load for monitoring active and passive state?

A4.3.4.2.2.5 T4 Mobility

The host is moving from one base station to another

?? Are there different mobility actions for different QoS classes? (The requirement is different. The answers to the following questions may differ for different QoS classes.)

?? How long is the network involved in hand over (have to be significantly less than the time a terminal may stay in an overlapping region between two base stations.

?? < 100 msec (is this a realistic figure?)

?? Will there be an interruption of the flow during hand over (e.g. due to temporary delay or packet loss).

?? < 1 or 2 packets (The goal shall be that hand over is not noticeable)

?? Is there a temporary change of QoS during hand over (shall be non or minimised)

?? What's the risk for packet loss due to hand over?

?? Is it possible to deal with handover per flow rather than per terminal? (An example is that a user may wish to move the voice path to GSM but not other paths. This example may however be an application issue and not a BRAIN issue?)

?? The terminal has to be "re-authenticated" when moving to a new base station. Is this an issue?

The criteria defined in D2.1 are:

A. Efficiency

?? Setup delay

?? Amount of signalling required to setup the QoS

?? Resource usage in the routers

?? Latency in re-establishing the QoS after handover

B. Scalability and Robustness

?? Scalability to large networks

?? Scalability to a large number of users

?? Complexity required in the network routers

?? Resistance to wireless errors

?? Resistance to link/node failures

?? Reliability of the requested QoS

C. Applicability and Ease of Deployment

?? Migration

?? Support of heterogeneous networks

?? Support in existing routers

?? Interaction with global mobility protocols

?? Interaction with local mobility protocols

?? Ability to react to changes in the network topology

?? Ability to support 'dumb' hosts

?? Ease of integration with accounting systems

?? Adaptability to different policies (e.g. fairness principles)

?? Interaction with changes in bandwidth

?? Interactions with general changes in the network

A goal is to minimise the number of routers that have brain specific coding. DiffServ routers are considered standard.

A4.4 Solutions to Weaknesses in the Base-Line Architecture

The following sections provide detailed descriptions of the proposed extensions to the baseline architecture.

A4.4.1 QoS Context Transfer

When the mobile changes BAR, state information about the mobile's QoS requirements needs to be transferred to the new BAR. Current handover schemes do not provide a mechanism via which this information can be transferred between BARs. QoS Context Transfer enables the exchange of network layer parameters between network nodes involved in a handover.

The following section outlines how link layer notifications of handover can be used to trigger the generation of context transfer messages.

A4.4.1.1 Link Layer Mobility and Context Transfer Protocol Coupling

The link layer necessarily sooner is aware of a hand over than the IP layer. The idea to make use of this early link layer hand over awareness to provide local triggers to the network layer is convincing and discussed not only with regard to QoS support (see section A5). Applied in the area of QoS, a link layer information to a BAR that a hand over is pending could be used to send a request for a context transfer by

the new BAR to the old BAR. Here it is not relevant how the context transfer works. This extension only describes a mechanism to start it.
The two features “link layer hand over indication” and context transfer protocol allow to admit new QoS resources between MN and BAN without requiring any signalling from the MN itself. In fact unless scarce resources require a QoS re-negotiation during hand over, the MN’s network layer may not become aware of the hand over at all. This solution provides a smart mechanism combining a seamless hand over with a reduction of IP layer air interface signalling. A more detailed description how this mechanism could be used may be found in section A4.4.1.

A4.4.2 Bandwidth Broker

In order to have a more accurate resource handling and flexibility in resource assignment, a Bandwidth Broker would be clearly needed. RFC 2998 also discusses the issue of dynamic admission control using a central “oracle”, but leaves the implementation open. A very interesting design of a Bandwidth Broker can be found in [A4.32]. The document presents a bandwidth manager for IEEE 802-style networks. The clients signal their resource requests with RSVP. The bandwidth manager intercepts the messages and makes admission control decisions according to resource availability in its network segment.

Since our QoS framework includes both RSVP- and DiffServ based resource sharing, we would need to upgrade the mentioned Bandwidth Broker scheme to understand DSCPs and to store resource utilisation for aggregate DiffServ classes. The Internet2 QBone Bandwidth Broker architecture [A4.34] provides an example how such a system could work.

Also other admission control criteria may need to be available. The Policy framework [A4.29] and COPS protocol [A4.30],[A4.31] can provide the necessary mechanisms. Benefits of having a Bandwidth Broker co-ordinate the network resource allocation are that the overall resource utilis ation becomes higher with more accuracy in the resource states of the whole network. The Bandwidth Broker however introduces a new point-of-failure; should the Bandwidth Broker crash, the resource allocations in the network will surely suffer. Also, having a Bandwidth Broker affects scalability and creates a possible bottleneck node.

The ISSLL Working Group of the IETF [A4.32] has done some work in this area. The bandwidth manager deals with the IntServ architecture over IEEE 802-style networks. This work has resulted in the development of the Subnet Bandwidth Manager (SBM) for shared or switched 802 LANs. SBM is a signalling protocol for RSVP-based admission control over IEEE 802-style networks. It provides a method for mapping an internet-level set up protocol such as RSVP onto IEEE 802-style networks. In particular, it describes the operation of RSVP-enabled hosts/routers and link layer devices (switches, bridges) to support reservation of LAN resources for RSVP-enabled data flows.

Basically, the SBM protocol performs at layer 2 the same functions as RSVP does at layer 3. In order to perform this, two primary components are required:

- a Bandwidth Allocator (BA) maintains state about allocation of resources on the subnet and performs admission control
- a Requestor Module (RM), in every end-station, performs the mapping between higher layer QoS protocol parameters and layer 2 priority levels

Two different SBM architectures are proposed, depending on the number of BAs per segment (Centralised architecture if there is only one BA, which must have some knowledge of layer 2 topology of the subnet, distributed otherwise.), as shown on

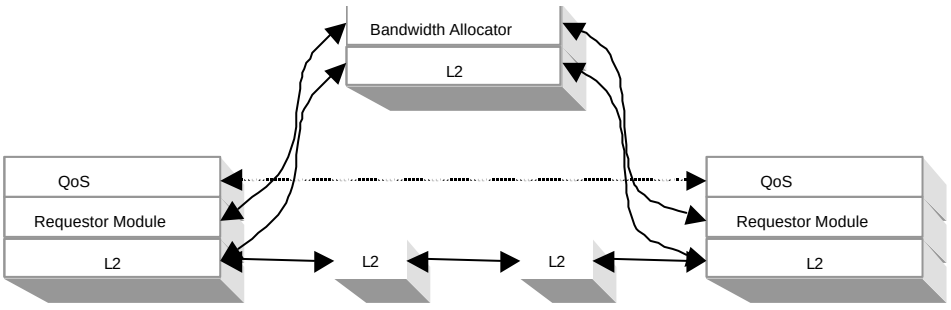


Figure A4-11: Centralised BA architecture

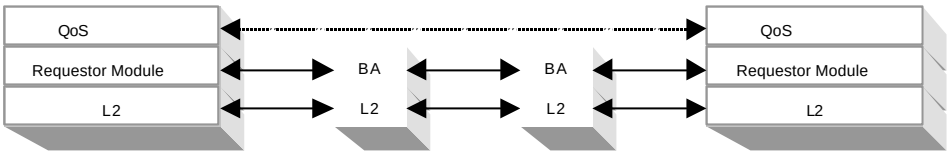


Figure A4-12: Distributed BA Architecture

In any case, the SBM protocol implies two types of communications between the different components :

- Communication between the higher layers and the RM (for the application to initiate, change or delete reservations, for the RM to inform the higher layers of a QoS unavailability, etc.)
- communication between the RM and the BA, or between BAs (a signalling mechanism similar to RSVP)

A4.4.2.1 The Internet2 QoS Broker

In the framework of the US Internet2 project currently a simple inter domain bandwidth broker protocol is specified. It is intended to support automated IP QoS resource reservations between peer bandwidth brokers. The signalling protocol is not based on any current IETF protocol and would support any type of DiffServ based service requests. The Internet2 bandwidth broker does not make any assumptions how a user would signal his request for quality differentiating services to its local bandwidth broker. As long as the service parameterisation required by the bandwidth broker for inter domain signalling can be derived from the local signalling information, a local provider is free to support whatever protocol he likes to with its users.

A bandwidth broker providing this kind of service could be useful to signal resource requirements to BAN adjacent provider networks supporting DiffServ based services only.

Internet2 intends to co-operate with IETF should the implementation of the protocol prove its usefulness. Contacts between Internet2 and the IETF Service Level Specification Work Group (should it come to existence) are already established.

A4.4.3 Coupling of hop-by-hop call admission with Current Micro-Mobility Mechanisms

The following investigation covers possible mechanisms by which the performance of reservation-based QoS, as defined in the Integrated Services architecture [A4.23], can be enhanced for the micro-mobile environment. Reservation-based QoS implicitly assumes that the route taken by a traffic stream across a network is reasonably stable for the duration of a reservation. The reservation is installed along the path using a QoS signalling protocol, the most widely adopted of which is RSVP. For simplicity, RSVP is used as an example protocol in the following discussion, but the concept can be extended to any other out of band soft state mechanism. When using IntServ with RSVP, changes to the path are handled by the soft-state nature of the architecture, and reservations are installed along the new path by periodic refresh messages. The installation of the reservation along the new route is not immediate, and the level of QoS received by a traffic flow can be temporarily reduced. In contrast, the routes in the mobile environment can be dynamic, changing every time the MN changes AR. Therefore, there is a need for fast re-establishment of paths and QoS reservations. If this is not supported, unacceptable disruption to the application traffic can occur every time the mobile node changes location.

In order to improve the behaviour of reservation-based QoS in the micro-mobile environment, the QoS and micro-mobility mechanisms can be coupled to ensure that reservations are installed as soon as possible, after a mobility event such as handover. In this study we present three levels of coupling over three different micro-mobility schemes. Here we present the key aspects of the three schemes relevant to this discussion, although a more deep classification can be found in section A3:

?? **Proxy agent architectures** [A4.35], [A4.36], [A4.37], [A4.38] tend to employ tunnels, either a single tunnel or a hierarchy of tunnels, to forward traffic to the CoA allocated to a MN. The tunnel-based micro-mobility mechanisms add scalability to RSVP because reservations can be aggregated onto a single trunk link between mobility agents, and support for QoS aware routing is possible because it will simply affect the route the tunnel takes across the network.

?? **The MANET-based scheme** considered in the following discussion uses MER-TORA [A4.39] to distribute the routing information within the network. After handover, a host-specific route is

inserted into the network, using a route update messages, to ensure that traffic travelling to the MN can be routed to its new location.

?? **Per-host forwarding schemes** use soft-state host-specific forwarding entries for each of the MNs within a domain. The entire domain has a special gateway that is the default route via which all nodes access the external network. Routing information is refreshed periodically, and updated immediately during handover to install the explicit route to the MNs new location.

The three scales of coupling presented for consideration are described on the following sections.

A4.4.3.1.1 De-coupled

In the de-coupled option, the QoS and micro-mobility mechanisms operate independently of each other and the QoS implementation is not dependent on a particular mobility mechanism. The QoS reservations are installed using RSVP signalling and IntServ control service parameters, and routing information is distributed using either standard or specialised micro-mobility routing protocols. Changes in network topology are handled by the soft-state nature of the reservations.

Potential problems with this approach occur when the MN hands over to a different AR and the path to and from the MN changes. A section of the old reservation, up to the point where the path to the old AR (OAR) and the new AR (NAR) intersect, is no longer valid because the traffic flows to and from the MN are now travelling via different network nodes. This node can be referred to as a crossover router, similar to the crossover router concept used in some micro-mobility schemes, and is illustrated in the Figure A4-13.

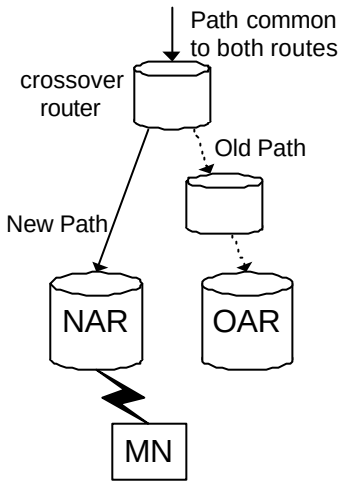


Figure A4-13: Concept of a Crossover Router

In order to provide the required QoS to the MN's traffic streams, reservations are required along new paths to and from the MN's new location. These are installed by the refresh mechanism used by RSVP to maintain the soft-state reservation information. The refresh messages are generated periodically, and in the mobile environment there will be a disruption to the agreed QoS during the interval between the MN moving location and the generation of a refresh message. If the refresh message is generated before the route to the MN's new location has been completely propagated throughout the network, the reservation will be made along an incorrect route and not corrected until the next refresh message. The reservation may even be refused if the resources are not available along the incorrect path or the router cannot route the data to the required destination. This will occur every time the MN moves AR, which may be many times during one RSVP session, and can lead to poor overall QoS for an application. In addition, the reservation along the old path cannot be explicitly removed, and must be left to timeout, which is not the most efficient use of network resources. These problems are common to all micro-mobility schemes.

A4.4.3.1.2 Loosely coupled

The loosely coupled approach uses mobility events to trigger the generation of RSVP messages, which distribute the QoS information along new paths across the network. The RSVP messages can be triggered as soon as the new routing information has been installed in the network. This has the effect of minimising the disruption to the application's traffic streams because there is a potentially shorter delay between handover and reservation set-up. It also avoids the problem of trying to install a reservation across the network before the routing update information has been propagated. The latency for installing

the reservation can also be reduced by localising the installation to the area of the network affected by the change in topology, i.e. between the crossover router and the NAR. The areas of the network affected by the topology change can have reservations installed across them almost immediately, instead of having to wait for the update to travel end-to-end, or for the correspondent node to generate a refresh message for reservations to the MN. In the case where the QoS must be re-negotiated, however, end-to-end signalling is required. The old reservation can be explicitly removed, freeing up unused resources immediately.

However, the loosely coupled approach requires additional complexity within the inter-mediate network nodes to support the interception and generation of RSVP messages when the router is acting as the crossover node. Another disadvantage is that bursts of RSVP signalling messages are generated after handover to install multiple reservations. This does not happen in the de-coupled case, because the reservation signalling messages are generated when refresh timers expire, not by the same triggering event.

In the **proxy agent architectures** the loosely coupled approach overcomes the problem of reservations being installed before valid routes to the MN are available by ensuring that the reservation is not installed until the registration information generated by the MN has propagated across the network. Reservations from the MN will not be installed until the acknowledgement of registration is received. This indicates that information concerning the CoA of the MN has been distributed in the network, and that a valid route to the MN's location is known. Reservations to the mobile can be created by the crossover mobility agent.

In **MANET based schemes,** the loosely approach associates the two mechanisms via triggering. For reservations from the MN, the receipt of a route update acknowledgement indicates that the explicit route to the MN's new location has been installed in the network, and causes the generation of the refresh messages to provide the fast re-establishment of the reservation. For reservations to the MN, the crossover router is responsible for generating the appropriate RSVP messages.

In **per-hop schemes** triggering is also used to perform the integration. For example, in HAWAII after a handover QoS signalling can be triggered once the new routing information has been distributed into the network and RSVP can make use of its routing interface to generate a Path Change Notification. The reservation is installed in the network as soon as the route to the MN is stable without having to wait until the next timeout to send QoS messages.

A4.4.3.1.3 Closely coupled

The closely coupled approach combines by using the same signalling mechanism to propagate the mobility and QoS information, either as an extension to the QoS/MM signalling protocol or via a unique QoS-routing protocol. This approach minimises the disruption to traffic streams after handover by ensuring that the reservation in place as possible after handover. However, instead of having to wait for an acknowledgement that the route to the MN is in place in the network, as with the loosely coupled approach, the QoS requirements for traffic flows travelling to the MN can be installed at the same time as the routing information. This avoids the problem of installing a reservation before valid routing information to the MN has propagated across the network, and also provides a means to install multiple reservations using one signalling message. This reduces the bursts of QoS signalling traffic sent across the network that occurs with the loosely coupled approach.

As with the loosely coupled strategy, the QoS reservation updates can be localised to the area affected by the topology change, unless end-to-end re-negotiation is required. The reservation along the old path can also be explicitly removed. However, the closely coupled approaches place requirements on the micro-mobility mechanisms to transparently carry opaque QoS information and additional complexity is required in the inter-mediate nodes. In some cases, additional micro-mobility messages are required to support this solution.

In the **proxy agents architectures**, this closely coupled version extends the loosely coupled strategy commented before for this scheme, with additions that support the opaque transport of QoS information in the registration messages. The addition of QoS information in the registration messages allows the MN to choose a mobility agent based on the available resources. This feature can provide some degree of traffic engineering within the network.

In the **MANET based scheme**, the closely coupled extends the loosely coupled solution so that the route update messages transparently carry opaque QoS information about traffic flows travelling towards the MN. The reservations are installed at the same time as the routing information, minimising the disruption to the traffic flows.

Finally, in **per-host schemes**, the closely coupled approach is very similar to that commented for MANET schemes. This integration can be performed easily because both mechanisms rely on soft-state signalling mechanisms based on path set-up and refresh messages. The suggested most suitable way to perform this integration is to extend the micro-mobility protocol to opaquely carry IntServ objects to distribute the QoS control information at the same time as the routing data.

A4.4.3.1.4 Comparison of Approaches

Coupling reservations with micro-mobility mechanisms allow reservation set-up delays to be minimised and packet loss reduced. Reservations along the new path can be installed faster because QoS messages can be generated as soon as the new route is established, reducing the disruption to the data flows. Also scalability and overhead are improved because a minor number of update messages are sent or they are localised to only the affected areas of the network.

Another advantage to coupling the two mechanisms is that it ensures that the request for a QoS reservation only occurs when there are valid routes to the MN in the network. Otherwise, the reservation will be installed along the incorrect route, and maybe rejected if the resources along that route are not available, or if the route to the required destination is unknown.

The closely coupled approach requires support from particular micro-mobility mechanisms so that the opaque QoS information can be conveyed across the network. This has the consequence that the QoS implementation will be specific to a particular micro-mobility mechanism, and extensions to the micro-mobility protocol may be needed to support the required functionality. However, the closely coupled approach maintains consistency between the reservation and the routing information within the network, and can reduce the amount of signalling required to set-up multiple reservations. The choice between whether to use the loosely coupled approach or the closely coupled approach is a trade-off between a QoS solution that is tied to a micro-mobility protocol and the performance advantage close coupling provides.

A4.4.4 Repairing RSVP local Path repair

RSVP nodes may implement local path repair mechanisms [A4.4]. These can be used to provide fast adaptation to local routing changes, such as those which may occur as a result of mobility. When a router (which we consider to include the mobile node) detects a change in the set of outgoing interfaces for a destination, RSVP should update the path state and send PATH refresh messages for all sessions to that destination. The delay between detecting a PATH change and sending a path change message is configurable and should be adjusted to give the mobility management mechanisms a chance to build the path. Once the new PATH message reaches a node that recognises that the message is a result of local path change, it should send a RESV message immediately - thus the end nodes need not know that the path has changed. Essentially, local path repair is using the detection of a routing change rather than a timer to initiate the soft state refresh messages. It enables quick re-establishment of QoS.

However, there is a problem with this if RSVP is used in hard state mode as it could result in "hanging reservations", indicated in Figure A4-14.

I propose one mechanism that could be used to avoid this is to use the data in a session to act as a refresh indicator for the session -an implicit signal that the reservation is still required. Although this will eventually result in the reservation state being cleared, these processes will be slow. This could cause problems in a bandwidth-limited environment. Therefore, the local path repair process can be further extended, as shown in figure below.

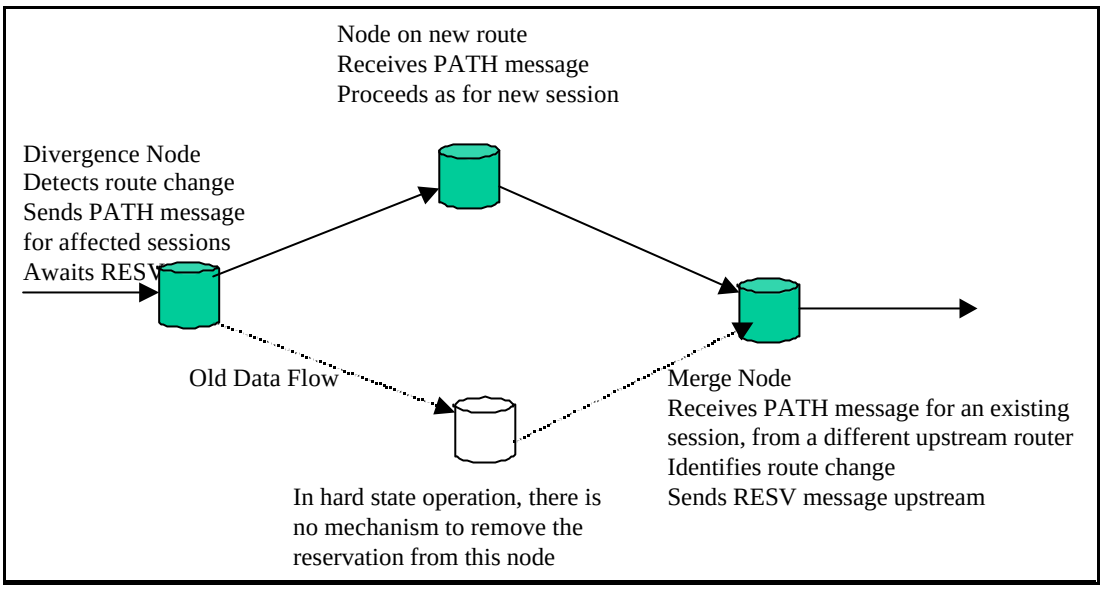


Figure A4-14: RSVP Path Repair

The simple solution to this problem is identified in the diagram above, where the merge node sends a RESV_TEAR message along the old path. This is terminated at the divergence node. The problem with this is that, since either the sending or receiving node can be assumed to be the divergence or merge node respectively, this system requires significant changes in the terminal RSVP implementation, and also involves sending an extra message unnecessarily over the wireless link. The message is unnecessary because the reservation at the wireless link can be assumed to have been cleared during the handover process, the problem is really a network problem not a terminal one, and the context transfer protocol means that the new access nodes have complete information about the required reservations. Therefore the solution depends upon the direction of data flow. Any network based merge node should send a RESV_TEAR message to the previous route. Any network based divergence node should send a PATH_TEAR node to the previous route. Access routers have a special role in terminating such messages.

Such messages must either be reliably delivered, or they must carry some marker to distinguish them from application initiated TEAR messages.

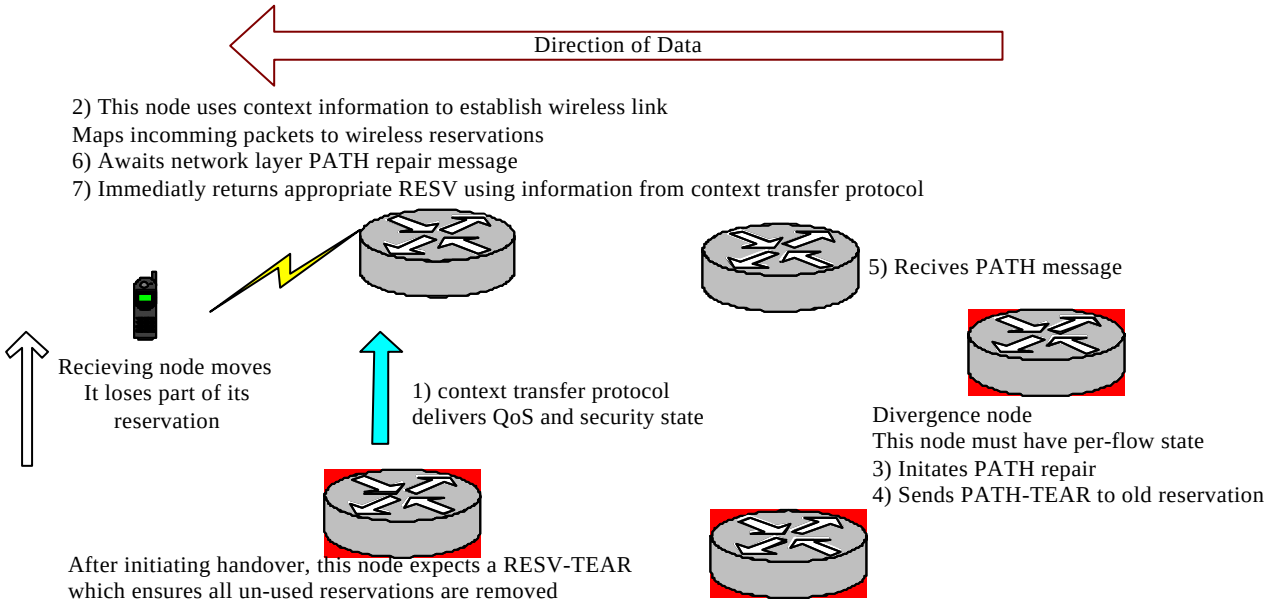


Figure A4-15: RSVP Messaging

A4.4.5 Bounded Delay Service

A4.4.5.1 Basic Operation

The bounded delay (BD) service has been proposed [A4.40], [A4.41] as a means to provide scalable, guaranteed real-time data transport within the Internet. It allows flows to have a guaranteed bandwidth and low, quantifiable queuing delay, whilst routing is simply based on traffic aggregates which are identified through the TOS marking.

For each output port, a node has a certain amount of bandwidth that is allocated to this service. Provided this bandwidth limit is not exceeded, all traffic using this service at that node has the same, guaranteed, worst-case routing delay. This worst-case delay is fixed for that port. All traffic for this service can then be scheduled using simple FIFO queuing algorithms.

Users of this service must request some bandwidth. This request is propagated through the network and resources are reserved at each node. For this, a robust signalling mechanism is needed. The user then marks the traffic with the required code-point, and must constrain their traffic to the agreed peak rate. To minimise the peak bandwidth required, a token bucket traffic shaper with the bucket depth equal to the maximum packet size may be used. In common with other DiffServ networks, traffic needs to be monitored ("policed") at entry to the network. However, other functions such as traffic shaping and marking are not necessarily required.[51]

The delay that a packet experiences through the network is the sum of the router delays and the transmission latency. It was identified before that real-time traffic has a delay budget of 200ms. For a transpacific transmission, the transmission latency will not be less than 80ms. Thus this leaves no more than 120ms available for router delays. Internet packets have a maximum number of hops – usually 30 - that they can be transmitted through before the packet is destroyed as undeliverable. This prevents circular routing problems. Thus the delay budget for router delays should be imagined as divided between 30 routers. It was further identified in section 3.6.2 that this delay budget should not be evenly divided between all elements of the network, as wireless networks need extra time to overcome the very high losses associated with transmission over wireless interfaces. The wireless transmitter typically needs 10 to 100 ms to achieve wireless transmission – this figure depends upon the type of wireless system used and the probability of successful transmission. Furthermore, both end terminals could have wireless interfaces. Thus, any single bounded delay node should set its worst case delay time up to 5 ms. Since, from [A4.40], we have:

?? the worst case delay for a node = (Number of BD flows * packet size for BD traffic + packet size of best effort traffic) / bandwidth of outgoing link

?? we can see that the higher values of delay are more suitable for low bandwidth links, where otherwise the maximum packet size (MTU) or number of bounded delay flows simultaneously supportable would need to be severely restricted.

In addition to the worst case delay bounds the authors in [A4.40] propose additional statistical delay bounds. Within the IntServ (and therefore ISSLL) Guaranteed Service, the worst case delay is always used in all call admission decisions. Particularly within the backbone network, where the statistical effects of a large number of flows become important, this leads to inefficient network. This is because bandwidth is reserved based upon the worst case delay, and to minimise this delay, large amounts of bandwidth must be reserved. However, this worst case delay will be very rarely experienced across the whole network path. The authors of [A4.40] define regions where different admission control strategies may be used, giving significant efficiency gains within backbone networks.

A4.4.5.2 Problems and Solutions

A4.4.5.2.1 Denial of service attack

A key problem with a network built as described in [A4.41], and originally in [A4.40] is its potential for denial of service attacks. This occurs because nodes simply maintain a bandwidth sum to determine admission control. When sessions close, this must be signalled so that the bandwidth sum can be adjusted, and bandwidth freed for another session. If these nodes do not maintain per-flow state, they cannot know if a session close request is valid. Thus a session close request may lead to the bandwidth sum being

[51] If the service is not standard throughout the network, then the Bounded Delay domain can be used as another DiffServ domain, which would have the usual DiffServ management functions at the entry (Ingress) and exit (Egress) to the network domain

reduced, which enables a new session to be started even if old sessions have not closed. Thus, if nodes do not keep per-flow state to validate signalling messages, they must be protected by an ingress router which polices the signalling messages as well as the data traffic.

This problem is expanded when mobility is supported. This is because correctly policed data may enter the BD network, and then be re-directed away from the original reservation by a route change. This is illustrated within the figure below, where is proposed that nodes within the micro-mobility zone keep per-flow state, as this helps manage QoS in the presence of mobility. This state is only held at the edges of the networks and is not adding to the complexity of the scheduling function, thus minimising the problems associated with storing per-flow state in routers.

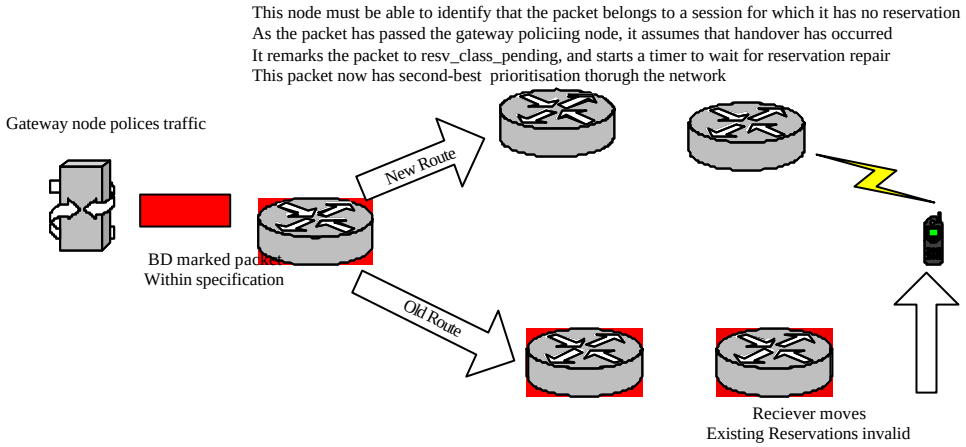


Figure A4-16: QoS and Mobility

A4.4.5.2.2 Load Balancing

Care must be taken with load balancing of the BD aggregate traffic. Traffic with QoS guarantees on a particular route must not be redirected to a new route, nor must such traffic be subject to load balancing [52], except at routers which maintain per-flow state and may therefore achieve a level of load balancing whilst ensuring that all traffic for a particular source-destination pair travels the same route.

A4.4.5.2.3 Handover Markings

This traffic marking is intended for reservation based traffic. When a handover occurs, there are two places where the required reservation will not be available – on the new data route within the network and in any temporary tunnels created between the old and new BAR. If no attempt is made to establish reservations in advance for this situation – and advance reservation are very difficult to do until the network path is known and is stabilised – then a static/dynamic guard band approach can be taken to help maintain QoS during handover. In this approach, each node reserves a portion of its available bandwidth to be used solely for traffic that enters the node as a result of handover. The amount of bandwidth reserve red for any particular DS class may be statically configured, or it adjusted dynamically based on knowledge of the state of the network. When reservation-marked traffic (identified through the DSCP) enters a node in which it has no reservation, it is assumed to be handover traffic. It can then be re-marked into the handover class DSCP.

A4.4.6 Simpler QoS Classes

Within both the IntServ/ISSLL and DiffServ solutions, a service has been targeted directly at delay sensitive applications. Neither of these is best suited to the wireless environment. The Guaranteed Service is not used for a number of reasons. Firstly it is using delay based admission. This makes centralised admission control difficult, particularly in a mobile environment where the state information available to the control node will always be stale. It also leads to complex scheduling mechanisms in routers. The service may be inefficient within the core network as the worst-case delay is always used in admission control, rather than making assumptions that this worst case delay will be rarely experienced within the core network where statistical effects can be significant. This inefficiency is exacerbated by the fact that

[52] Not strictly true, if certain conditions are met about the load balancing routes, then load balancing can exist.

to minimise delays, larger amounts of bandwidth will be reserved than may strictly be necessary. The guaranteed service promises not to discard packets once within the router. It is not clear that a wireless router could make this guarantee if significant fluctuations occurred in the wireless environment. Finally, billing is difficult for delay based admission services. The DiffServ Expedited Forwarding (EF) Service is not used as no single definition of this service now exists. It is currently the source of much debate within the IETF, with two separate attempts to develop more precise definitions. It is my understanding that one of the proposals is very similar to the service definition used here, but the other, more popular, proposal is aiming to define the EF service in line with the IntServ guaranteed service for operation in the ISSLL architecture.

A4.4.7 Mobility Enhanced QoS Parameters - Generic IP QoS signalling interfaces in a mobile environment

This section introduces a number of generic interfaces and parameters required to support mobility aware IP QoS mechanisms. A general assumption made here is that any reservation of resources is unidirectional. This assumption eases interaction with IP routing protocols, which is thus limited to mobility management. Bi-directional QoS reservations may require additional IP parameters or interfaces. Bi-directional QoS reservations are left for further studies.

Two separate problems have to be solved by a QoS mechanism in a mobile environment: set-up, maintenance and release of QoS resources. This is called ”Static QoS resource management” in the following. A special case occurs during hand over, when parts of the mobile network try to re-route an already accepted QoS reservation to some new systems within the same network. This is called ”QoS management during hand-over” in the following.

The following includes a minimum number of parameters whose support is required to operate mobility aware QoS services. An implementable solution may specify more parameters. None of these additional parameters is however generic (i.e. is required by all solutions).

A4.4.7.1 Static QoS resource management

It is a general requirement that all messages referring to a particular reservation pass all and the same admission points while the pass the Internet from one end node to the corresponding one *and back*. Emphasis must be put on this, as a decentralized admission control must solve the problem of asymmetric IP routing. A more serious issue may arise, if multihomed networks are transited. In this case, also a centralized admission process may fail if not designed carefully (the problem arises if a complete domain is shunned by a communication in the reverse direction). A solution to the problem is to require every admission control point to insert its own IP address into the initial Resource Set Up message. The next downstream admission control point will have to store this address and then insert its own. By using this preceding admission control point information to route upstream messages, any signalling packets travelling upstream will pass the same admission control points as the downstream messages. Whether or not this is a generic parameter remains for further study.

QoS Resource Set Up messages are created by an MN or CN. Contents of Resource Set Up messages must be interpreted at network edges (BAN and BMG in the case of BRAIN).

Message: Resource Set Up

Status: Mandatory

Mandatory Parameters

Reservation ID

Service ID

Negotiation Flags

As long as a reservation is in active state, the Service ID must be stored by the admission and resource control elements of each domain passed by an individual domain. No assumption is made whether this is done in hard- or soft state, centralized or distributed fashion.

Acceptance or rejection of a Resource Set Up is indicated by a (No) Set Up Acknowledge message, created by the corresponding end system. The Negotiation Flags indicate whether the Resource Set Up message is a response to an indication that the original reservation must be re-negotiated to a lesser amount of resources if it should pass.

Message: (No) Set Up Acknowledge

Status: Mandatory

Mandatory Parameters

Reservation ID

Acceptance Indicator

The originator of the Negotiation message may be any admission control point along a path, the CN or the MN. The Negotiation must refer to the same service as requested by the resource Set Up. The basic operation would be to forward the Negotiation message to the originating terminal. This then replies with a Set Up containing the same Reservation ID and a modified Service ID (or a Release message, if the offered service parameters are not acceptable). It contains the following parameters:

Message: Negotiation

Status: Mandatory

Mandatory Parameters

Reservation ID

Service ID

Negotiation Flags

Note that a negotiation process could be started already during the Resource Set Up process. A transit admission control point may recognize that it can't provide the requested resources and it may insert a negotiation offer by changing the service ID information. This must be indicated by a flag. If the following downstream domains and the CN accept the changed Resource Set-Up, The Set Up Acknowledge message must include the changed Service ID and a Flag indicating the a negotiation took place. It's then up to the MN to accept the reservation or answer by a Release. The definition of a negotiation process as given here is only an example.

The Release message removes the reserved resources. The release message may be sent by the terminals or by any intermediate admission control point. If sent by an intermediate admission control point, it must be directed upstream as well as downstream.

Message: Release

Status: Mandatory

Mandatory Parameters

Reservation ID

Location ID

Cause ID

A4.4.7.1.1 QoS management during Hand-Over

This section explains the mobility centric interfaces required. It may sound amazing, but a single dialogue may suffice:

The Hand-Over Indication is sent by the new access router to the admission control point of the terminal in the process of hand-over. In the case of BRAIN, it is the new BAR.

Message: Hand-Over Indication

Status: Mandatory

Mandatory Parameters

Reservation ID

Note that the Hand-Over Indication may just be an internal event within the new access router (e.g. if a soft state network layer mechanism like legacy IntServ/RSVP is operated).

The Hand-Over Resource Information must be sent to the new access router whenever Hand-Over Indication state is active for a terminal handing over. The Hand-Over Resource Information may be sent from the (old) admission control point of the terminal after it received a Hand-Over Indication message form the new access router. It may as well be sent form the terminal in the process of hand-over itself.

The information provided by the Hand-Over Resource Information will allow an admission control decision at the new access router.

Message: Hand-Over Resource Information

Status: Mandatory

Mandatory Parameters

Reservation ID

Service ID

The hand-over support definition given here is flexible enough to support a variety of different implementations. A pure soft state model like legacy IntServ/RSVP may suffice as well as a pure hard state model based on a centralized Bandwidth Broker (which may be seen as two extremes).

A4.4.7.1.2 Parameter Information

This section explains which information must be transported by the parameters introduced above.

Reservation ID:	A set of parameters identifying a resource reservation in a non-ambiguous way (like source and destination IP addresses and/or port numbers, application and DiffServ code point and/or an arbitrary but unique identifier).
Service ID:	A set of parameters defining a service (like the PHB ID) and the parameterisation of it (like the IntServ TSpec, Flowspec, Adspec and so on parameters).
Negotiation Flags:	A set of flags indicating that the Service ID is modified against the currently active reservation.
Location ID:	Location of the system causing a release (e.g. IP address)
Cause ID:	The cause for a release of a reservation (this may be a standard reservation termination, lack of resources, or an error condition and so on).

Table A4-1: Parameter Information

This chapter suggests an initial solution for a BRAIN QoS mechanism. This mechanism is analysed against the requirements and the evaluation criteria for a BRAIN QoS mechanism. The weak points identified by this analysis build the basic idea of several suggested enhancements to existing standards. To clarify the consequences of following a specific enhancement as suggested, also these proposals are analysed against the requirements and evaluation criteria for a BRAIN QoS mechanism. Finally, a conclusion describes the added value as well as the traded benefits of the suggested enhancements.

A4.4.8 Optimised RSVP

The current IETF IntServ/RSVP signalling mechanism is intended to support a many to many communication. Between five and seven QoS parameters have to be specified and partially operated by each transited RSVP aware router. RSVP however can be used to carry non-IntServ QoS objects . Thus it would be possible to develop a solution that tries to optimise the use of t IETF RSVP signalling standards and architecture for usage in a wireless mobile environment. A central part of this idea is to enhance and change existing IETF specifications.

A4.4.8.1 Analysis against the D2.2 requirements for a BRAIN QoS mechanism

If an optimised RSVP is used, it will not support IntServ (which may be implemented in parallel). Support of DiffServ is a central idea . The BMG should provide an interworking function to allow QoS reservations to and from legacy IntServ nodes without using IntServ within the BAN or at the MN. As long as the RSVP optimisations are not standardized, the MN, the BAR and the BMG have to support BRAIN specific protocols and interfaces. Since a BRAIN compliant network domain must supports MNs using legacy IntServ, the BAR would have to provide the interworking (or in fact, support IntServ). The replacement of the IntServ QoS description objects by a mobility-optimised objects is the core idea of this proposal For benefits of this approach, refer to the evaluation section.

This approach is based on a coupling of link layer mobility management and the QoS resource administration in the case of hand-over. Enabling seamless hand-over is an important design criteria

A4.4.8.2 Analysis against the D2.2 evaluation criteria for a BRAIN QoS mechanism

Optimised RSVP supports the signalling for QoS resources in a generic way. It is assumed that globally well known services only make use of the mechanisms defined to reserve release QoS resources. The globally well-known service definitions themselves are out of the scope of this discussion.

The amount and contents of MN originated QoS signalling messages are minimized. No assumption is made on the usage of a hard- or soft state signalling. This is possible by de-coupling hand-over signalling from MN network layer signalling. The proposed solution defines a QoS context transfer between old BAR and new BAR in the case of hand-over. The MN is not expected to signal any information during hand-over at all. The amount and contents of signalling within the BAN to support hand-over is minimized too.

By coupling the QoS support during hand-over to link layer mobility indications, seamless hand over is supported. packet losses are not expected during a hand-over, and additional packet delays are not introduced. In fact, if the new BAR is able to provide the QoS resources required, the MN's IP layer should not notice that a hand over occurred at all (seen from QoS perspective only).

If the IP layer mobility protocol is not based on tunnelling mechanisms, optimised RSVP may be deployed and operated completely independent the IP layer mobility protocol. An IP layer mobility mechanism based on tunnels will require interworking procedures (i.e. set up and release BAN internal tunnels with the appropriate QoS during hand over).

The QoS-resource set up delay depends on the network admission control architecture A central BAN resource administration system with a lack of resilience, and long or congested links to the BARs and BMGs may cause serious set up delays. Set-up delay is a network design issue, not a QoS signalling protocol issue.

Optimised RSVP is simple and scales as well as DiffServ based core IP networks do. A large number of users shouldn't be major issue as the minimization of QoS related signalling is an important design criterion. Complexity in BAR and BMG routers is medium to high. This depends on the functionalities provided by routers (if a router provides signalling, admission control, and interworking, the complexity must be regarded as high). As in ordinary RSVP, only uni-directional reservations are supported. There is no intent to change IETF protocols to support bi-directional reservations. It is not planned to support reservations from CN to MN if the CN is unable to signal its requirements for QoS to the BAN.

Legacy signalling protocols like IntServ/RSVP should preferably be supported by BAR/BMG interworking. This allows the BAN to operate with all the benefits brought by optimised RSVP without significant disturbance of the service provided.

Whilst optimised RSVP requires changes to the existing RSVP protocol. RSVP isn't completely re-invented. Most of RSVPs procedures, messages and parameters will be used (though the parameter contents, as defined within IntServ, will have to be re-defined).

If optimised RSVP is not standardised and interworking is required, then the end-to-end Internet principle is violated. Strict layering is respected (with the exception of using layer two mobility management to trigger network layer signalling).

A4.4.8.3 Conclusion

Optimised RSVP suggests enhancements to the IETF RSVP protocol making it suitable for application in a mobile wireless network. While seamless hand-over and minimization of the network layer signalling load are very desirable, the drawbacks of having a BRAIN specific QoS signalling protocol are very large. Hence implementation of a such a QoS signalling mechanism is deemed to be reasonable only if it is brought on the standards track. As other enhancements to the general BRAIN QoS solution are expected to require changes to existing IETF protocols too, decisions regarding QoS related inputs to standards bodies are suggested for the MIND project.

A4.4.9 Local BAN signalling Protocol

One of the identified weaknesses of the baseline architecture is that it assumes the presence of end-to-end QoS signalling e.g. RSVP, that traverses the entire network between the communicating nodes. It is desirable to make BRAIN compatible with existing IETF standards but at the same time not assume any functionality in external networks and hosts.

If the BAN is viewed as a potential bottleneck of network resources, it may be advantageous to provide additional services within the BAN that may not be necessary end-to-end. For example, it might be that correspondent node in the external network does not support RSVP, so there is no point in propagating the message outside the BAN. Alternatively, the BAN may be connected to a provisioned core. In this situation, RSVP is beneficial only within the administrative domain, i.e. the BAN, in which it has been enabled.

The benefits of providing a local BAN QoS mechanism is that mobility enhanced extensions can be supported easily without affecting the end-to-end QoS signalling mechanisms. For example, it may be desirable to support different QoS classes in the local end and in the remote end. The BAN can support mobility enhanced QoS classes that are transparent to the correspondent node and external network. Where local BAN signalling is used, this is simple because the end-to-end QoS signalling is used as an independent overlay to the BAN internal QoS signalling. When standard RSVP is used, with proxying when end-to-end signalling is not required, the mobility enhanced parameters, if used, will need to be inserted and removed from the messages at ingress and egress of the BAN. While the RSVP proxy approach is valid, it is easier to support specialised parameters with two levels of signalling. The following figure provides an overview of the different levels of signalling that can be used for session establishment.

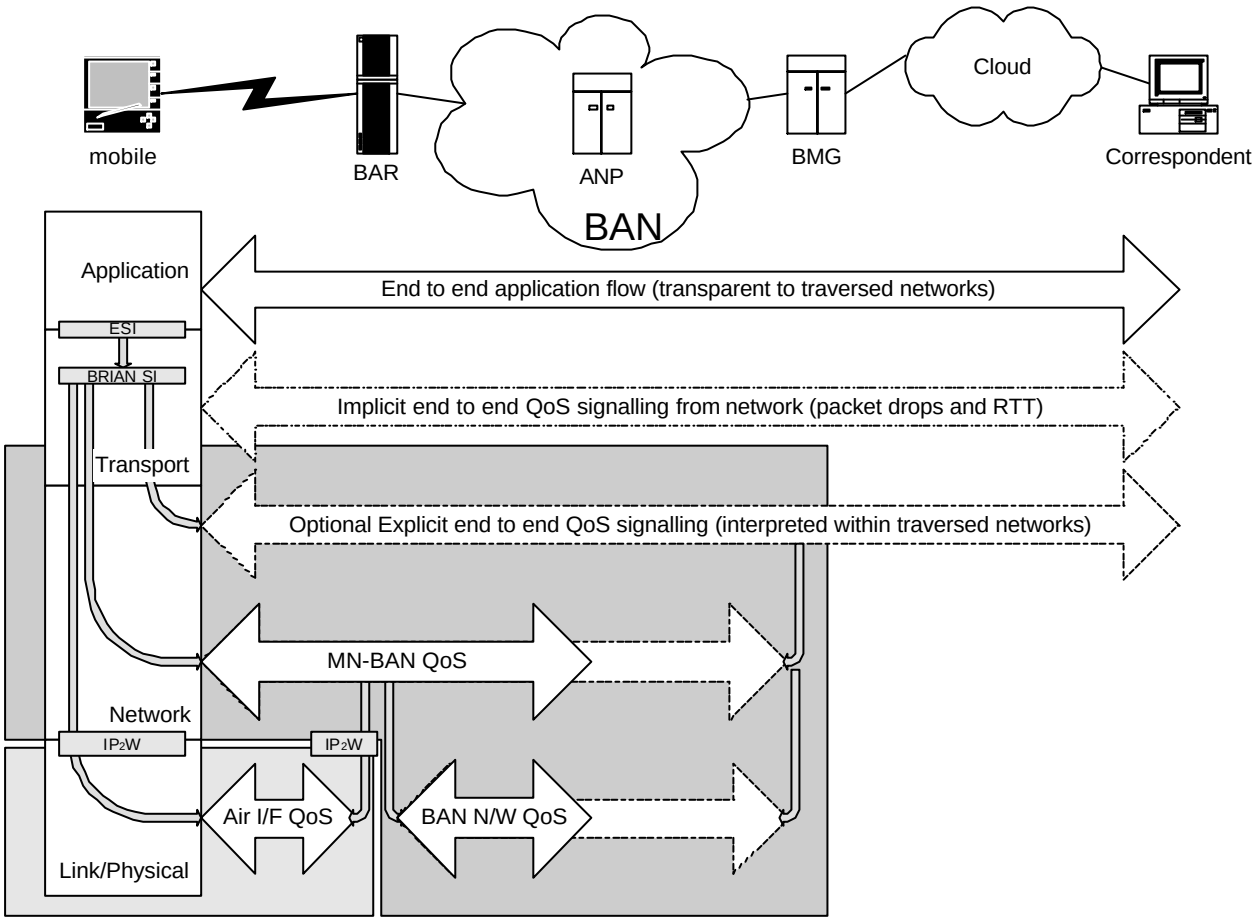


Figure A4-17: Session Establishment

The end-to-end signalling exchanges pass transparently through the BAN between the correspondent and the mobile. For both uplink and downlink flows, the mobile has access to the QoS requirements either directly from the application, or from application layer signalling. The mobile can use this information to ensure that the required resources in the BAN are made available to the traffic flows, and requires a mechanism by which it can reserve the local BAN resources.

If the ANP is located towards the BMG, then the BMG can be considered as part of the external IP network, and the ANP can become a logical place to locate the RSVP proxy, or to terminate the local signalling. The ANP for a mobile is a fixed point in the network through which the downlink traffic must pass while the mobile maintains the same CCoA. If the ANP is located toward the BAR, then the

location of the proxy/local BAN signalling termination point is not so obvious, but will probably be in the BMG. Determining which BMG the downlink traffic will pass through is non-trivial.

QoS reservations for the downlink traffic must originate from the ANP/BMG to ensure that the reservation is installed along the correct route. Therefore the mobile must have some means by which it can signal the QoS requirements for the downlink traffic flow to the correct anchor point. This mechanism can be based on either RSVP proxies or local BAN signalling. For example, the path update messages generated during handover of the mobile can be used to convey the QoS requirements for the sessions to the anchor point, which is then responsible for signalling the QoS requirements to the intermediate routers using, for example, RSVP.

When local BAN signalling is provided using extensions to an existing QoS mechanism, care must be taken to ensure that the mobile does not have to support two versions of the same protocol, for example, one version of RSVP for end-to-end signalling, and one version for local BAN signalling. The RSVP proxy approach solves this problem by extending standard RSVP to cope with both types of signalling. The tightly coupled QoS signalling protocol is only used between BAR and BMG, so the mobile is ignorant of this protocol. However, there must be some mechanism by which the mobile can inform the BAR of its QoS requirements in order to trigger this signalling. This could be by interpretation of the end-to-end RSVP messages, with the BAR stripping out any specialised parameters. Alternatively, the mobile could use an alternative signalling mechanism over the air interface, such as the combined L2/L3 signalling protocol (section A6). However, this protocol will not be supported by generic IP hosts.

When bcal BAN signalling is supported, the mobile user also has the option of signalling filtering information to the BAN to prevent malicious use of the limited resources

A4.4.10 RSVP Proxies

In the absence of end-to-end support for QoS, the user at a MT may choose to request resource in the BAN, for example, in order to get better QoS on the wireless link. Presently we can identify two ways to signal QoS requirements to an access network. One way is to use DiffServ Code Points (DSCP), the other way is to use RSVP.

With DiffServ the mobile node can mark the upstream packets if it knows the proper DSCP values. For the downstream we have to instruct the gateway node to mark the incoming packets with a certain DSCP. This can be accomplished by defining default values for different micro flows in the SLA negotiated between the client and the ISP. A second method would be to use a Bandwidth Broker [A4.49] that would dynamically give the proper code point on a per-flow basis: when the first packet of a flow arrives, the gateway would request the proper code point from the Bandwidth Broker and cache the information (keep a soft state) for future packets belonging to the same flow. A third way would be to define a protocol that the mobile node could use for dynamically adjusting the SLA stored at the gateway in order to override some default mappings.

The other mechanism for signaling QoS needs to the access network would be through RSVP. For upstream reservations, the mobile node would send the PATH message to the gateway, which would return the RESV message and setup the reservations. The gateway would act as an RSVP proxy [A4.50]. Setting a reservation for the downlink direction is however not as straightforward, since the downlink reservation needs to be initiated by the RSVP proxy. We would need some way to trigger the proxy to initiate the RSVP signaling for the downlink flow.

These mechanisms therefore do not solve the whole problem. The DiffServ mechanisms don't allow for explicit resource reservations and are less flexible for giving changing treatment to incoming flows. The problem with the RSVP proxy approach is that the proxy cannot automatically distinguish reservations that would be answered by the correspondent node and reservations that would require interception. Additionally, the RSVP proxy needs some way to know when to allocate resources for incoming flows.

The proposed scheme is based on the RSVP proxy proposal (a) and the RSVP local repair mechanism (b). We also need a way to differentiate reservations that are internal to the access network. We suggest using one bit of the four flag bits in the RSVP common header for this purpose (c). We name the flag RSVP Proxy Flag (RPF). The enhanced RSVP proxy that will be the partner for the local signaling is named the Correspondent RSVP Proxy server (CRP). We also add a new message type called "Proxy PATH" message.

When a mobile node wants to reserve resources in the local network, it uses the RPF flag to indicate a local reservation. The structure of the RSVP message follows the standard, even the intended receiver is set to be the host that the mobile node is communicating with. The CRP that intercepts the RSVP message

will notice that the flag was set, does not forward the message further and responds according to the following description.

A4.4.10.1 Upstream transfers

Setting upstream reservations is most straightforward and follows the RSVP Proxy functionality. The mobile node sends the usual PATH message, destined to the correspondent node it wants to send prioritized data, and sets the RPF. When the CRP receives the PATH message, it notes that the reservation is meant to stay within the access network and responds with a RESV message back to the MN. The RESV message reserves the resources if available.

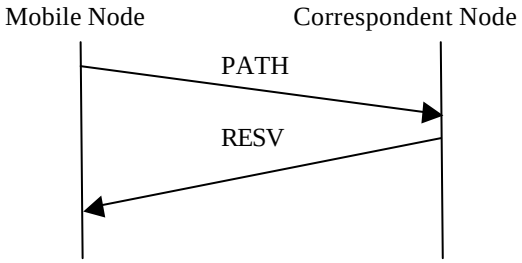


Figure A4-18: Upstream Signalling

A4.4.10.2 Downstream transfers

For downstream transfers we need a way to signal the CRP to initiate the RSVP reservation setup for the downstream on behalf of the correspondent node. To do this, the mobile node sends the Proxy PATH message with the RPF set and with the destination of the correspondent node. The Proxy PATH message is identical to a standard PATH message apart from the message type field. When the CRP receives this message, it notes that the message is meant to stay within the access network. The message type indicates that the CRP should initiate an RSVP reservation (for the downstream direction) and use the information in the Proxy PATH message to fill the field in the new PATH message. The CRP then sends the new PATH message with the RPF flag set to the mobile node. It sets the sender IP address to be the original destination. The mobile node receives this message and responds with a RESV message, that has the RPF flag set. This reserves the resources within the access network for the downstream.

All the other RSVP functionality work in the standard way, including the local repair mechanism and reservation tear down. All related messages must have the RPF set in order to keep the signaling within the access network. Intermediate RSVP routers between the mobile node and CRP should forward the Proxy PATH message as an ordinary IP packet.

An important functionality in each CRP is how will it know when the downstream reservation is not needed, for example, when the specific flow that had a reservation set up has ended and the mobile is out of coverage to explicitly indicate that the reservation can be removed. We suggest that the CRP together with the MN will keep the reservation in place (by sending the standard period refresh messages) if the indicated traffic is flowing throuhg the CRP. After a suitable timeout period, the CRP can release the resources

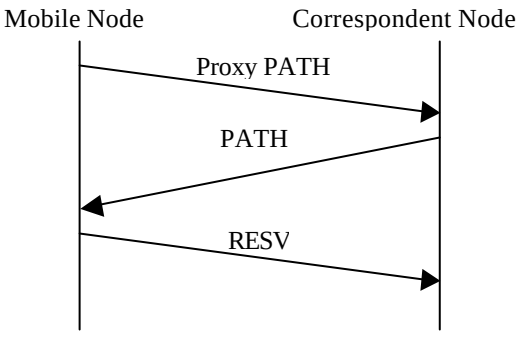


Figure A4-19: Downlink Signalling

The scheme also allows RSVP to be used to signal DiffServ Code Points in the BAN using the RSVP DCLASS object [A4.51]. The mobile node can use the DClassobject to instruct the CRP to mark incoming traffic with certain DiffServ Code Points to trigger different forwarding behavior within the access network. The CRP needs to have some table for mapping the RSVP request to DiffServ classes. Thus the mechanism can also be used to give relative priority to some flows, without explicit resource reservations.

A4.4.10.3 Usage Scenarios

An example use case for the proposed signaling mechanism includes a multimedia application that uses SIP [A4.52] to initiate a session and indicate that the other end does not support DiffServ of IntServ. From that information the mobile node could still setup local resources to enhance the quality of the communication.

In other scenarios, the mobile node could use the DCLASS objec for setting priorities for incoming flows, for example, medium priority to all web-browsing using HTTP and low priority to FTP-traffic.

A further possibility would be to reserve some bandwidth, for example 256 kbits, for all the mobile nodes communications on both directions. This would be seen from the mobile node as a circuit-switched connection to the local network. All traffic coming from the mobile node or destined to the mobile node from the external network would be policed against the reserved bandwidth. The mobile node could use by itself some prioritized queueing mechanism to allocate the reserved bandwidth to the active applications.

The Proxy PATH message could potentially be used in mobile networks to initiate a local repair on behalf on a mobile node that is receiving some flow. The standard RSVP processing defines that RSVP RESV messages cannot perform the local repair. Instead, when the mobile node has moved, it will need to wait until a PATH message is sent downstream that will refresh the reservation states on the new route.

When the mobile node changes its point of attachment to the network it should send the Proxy PATH message immediately after the handover. The message is forwarded through the intermediate RSVP routers until it finds the cross-over RSVP router that has the reservation for the mobile node stored on different interface. The message would then instruct the cross-over router to initiate a local repair by sending the needed PATH message.

The RPF must be set if the reservation was set for the local network. This will prevent the Proxy PATH message to be routed out of the local network; the cross-over router may still be located between the CRP and the mobile node and will therefore respond to the message. The closest of the CRP and cross-over router will thus respond to the routing change.

A4.4.10.4 Problems with the Approach

The main problem with the proposed mechanism is the decision about when to reserve only local resources and when to reserve resources on the full end-to-end path. This is mainly related to upstream flows, since downstream flows will be identified as either belonging to correspondent node initiated RSVP reservations or not.

When the mobile node reserves only local resources, the resulting end-to-end service may, however, not be as good as needed. This is because in the lack of end-to-end signaling, the network path between the CRP and the correspondent node may be congested or configured in such a way that the resulting service is poor.

A more complicated problem is related to network routing. It is possible that the routing between the mobile node and the correspondent node differs between the directions; the upstream flow will go through a different CRP than the downstream traffic. Thus, if the mobile node would want to signal some QoS for the downlink flow, the state might be stored at a wrong CRP. This problem is however more theoretical than an actual key problem.

The problem with different routing paths can be solved with multicast reservations. When the mobile sends the Proxy PATH message, it would use as the receiver address a known local multicast address that all the BAN CRP nodes belong to. Thus, every CRP (which should not be more than a few) will get the Proxy PATH and responds with the PATH message. The mobile will receive the PATHs and respond with a RESV, thus reserving resources in each direction. Since the flow the reservation was meant for will arrive through one of those CRPs, the other CRPs can just let the reservation timeout, RSVP is soft state.

A4.5 Discussion Topics

The following section highlights some of the discussion issues that were raised during the project. No all have been satisfactorily resolved, but are included for completeness.

A4.5.1 Protection of the mobile terminal

The above section discussed the policing functions, included as standard within a DiffServ network that are intended to protect the network from abuse. In addition, both application layer servers and the network itself can provide protection to the mobile terminal. This protection is essentially about preventing "SPAM". This is important, both because mobile terminals have limited resources and also because the mobile user is often responsible for paying for any data transmitted across the wireless network.

A4.5.1.1 Application Layer Proxies

These include SIP proxies, mail servers, mobile IP home agents. These proxies have rules established under user control, typically as preferences when you first register…. I think that within the IETF community, these would be preferred over protection based on IP source address as they can do more intelligent protection – based on content/context. Think of your wife, in hospital about to have her baby, and the hospital IP address is not a recognised address.

The application layer protection is probably more widely used today.

A4.5.1.2 Protection of Mobile Terminal by the BRAIN network

In order to protect the MN and its scare resources over the radio interface it would be desirable if the MN dynamically could specify filtering parameters to be used in the BAN. If the filtering is to be done at the BMG or BS (or somewhere else) is an implementation issue. However the protocol to be used have to be standardised. An example is that the user specifies that only data with a specified source address shall be forwarded to the user. A possibility might be to enhance the authentication protocol, but this is for further study.

If filtering is performed at the BMG, need to know the IP address of the relevant BMG. This would be a problem if there are many gateway nodes.

A4.5.2 Candidate Handover Node Selection

Problem Addressed:	Seamless handover
Scope of Impact:	Confined to mobility-aware network nodes
Status:	known solution in literature, not yet studied within IETF

Current handover mechanisms do not allow the selection of the new BAR based on anything other than signal strength, and so do not take into account resource availability across the air interface or within the access network. Candidate handover node selection allows the new BAR, or other nodes such as an anchor point, to be selected according to the mobile's QoS requirements.

When the MN is moving, the old BAR compiles a list of candidate BARs to which the MN could handover, and interrogates them to determine their current resource availability. The new BAR that most closely meets the mobile's QoS requirements is selected from this list of candidate BARs. The QoS information required to make this decision is exchanged during a context transfer phase of the handover procedure. The anchor point with which the MN registers can also be selected based on QoS criteria to ensure that sufficient resources are available.

Selection of candidate hand-over nodes based on QoS requirements requires support from the micro-mobility mechanism to allow the transfer and negotiation of QoS parameters, but is limited to the handover negotiation signalling messages exchanged between the mobile and the network.

A4.5.3 QoS and the BCMP

The following section shows how QoS sessions can be created, maintained and released across the BAN as the location of the mobile changes when using the proposed BRAIN micro-mobility mechanism. In this scheme the mobile registers its presence in a BAN with an anchor point, which is then responsible for forwarding traffic destined for the mobile to the BAR to which the mobile is currently attached. Tunnels are used between the anchor point and the BAR to forward traffic to the mobile. Path updates messages are send to the anchor point to keep it informed of the whereabouts of the MN. More information about

the protocol can be found in section A3.5. In the following discussion, RSVP is used as an example of an out-of-band QoS signalling mechanism, but the same process can be applied to similar QoS mechanisms.

The intermediate routers between the anchor point and the BAR can be RSVP-aware, DiffServ capable or a combination of both. If the RSVP messages are interpreted hop-by-hop on a per-flow basis, the fine-grained control over the QoS the application receives is possible. However, there is also a high overhead in terms of signalling and processing associated with this approach. Aggregating traffic flows onto one reservation that is interpreted hop-by-hop across the network can reduce this overhead, and DiffServ can be used to provide differentiated QoS for the flows within the aggregate. If the some or all of the intermediate routers do not support RSVP signalling, then DiffServ is used to provision QoS between the RSVP-aware routers. In this scenario, RSVP can be used to perform admission control at various points within the network.

Two different deployment scenarios are considered. In scenario A, the anchor point learns about the QoS requirements for downlink traffic flows from the end-to-end explicit QoS signalling (RSVP). In scenario B, the QoS for the downlink flows are signalled to the anchor point by the mobile, and the PATH messages are generated as soon as the path update is received at the anchor point. Scenario B can provision resources across the BAN in the absence of end-to-end RSVP signalling, which may be desirable if the resources within the BAN are limited compared to resources in the external network. The mobile may have information about the QoS requirements for a traffic flow from application layer signalling.

It may be desirable to introduce mobility-enhanced parameters for QoS within the BAN. These mobility-enhanced parameters can include information such as time periods for QoS measurements such that QoS violations are not triggered unnecessarily, and loss profiles and bit-error rates that are acceptable to the application. In scenario A, these mobility enhanced parameters would need to be signalled to the BAN in the end-to-end messages, but must be removed before being signalled into the external networks. Alternatively, the parameters are only interpreted by mobility aware entities, and are ignored by other RSVP aware routers. In scenario B, these enhanced parameters can be included in the PATH and RESV message used to reserve the resources in the BAN, which are independent from the end-to-end messages.

The following sections provide an overview of how QoS sessions can be created and maintained in a BAN that supports the proposed BRAIN micro-mobility protocol.

A4.5.3.1 Session Creation

Within the BAN, it is the responsibility of the anchor point to reserve resources for the downlink flows. The anchor point establishes the tunnel for the downlink flows and can ensure that the reservation is set up along the correct downlink path. If reverse tunnelling is used between the anchor point and the BAR, it is possible that the anchor point can set-up a reservation for both directions of a bi-directional traffic flow, such as a voice call. Otherwise, reservations for the uplink traffic flows can be installed using standard RSVP.

In an end-to-end signalling scenario the anchor point has to be able to process the QoS signalling in order to establish the needed reservation across the tunnel to the mobile. If RSVP is used then the anchor point has to be RSVP-enabled and may also be responsible for introducing specialised parameters into the reservation specification.

In the absence of end-to-end signalling (scenario B) the anchor point must be informed of the QoS for the downlink flows by the mobile. This can be signalled to the anchor using standard RSVP or in a more coupled solution can be included in the path update messages sent to the anchor point

The anchor point can aggregate the individual flows along the tunnels and make intelligent routing and traffic engineering decisions. Since the intermediate routing between the anchor points and BARs is based on standard IP routing protocols, current IETF standards for QoS routing and traffic engineering can be deployed.

Admission control for traffic flows should be carried out for traffic flows traversing the BAN. For uplink traffic flows, the BAR carries out the admission control. For downlink flows, the BMG or the anchor point can perform admission control. In both cases, RSVP-aware intermediate routers may also perform admission control.

A4.5.3.2 Session Maintenance

Refresh messages must be generated periodically to maintain the reservation across the network. If the flows are aggregated within the tunnels, the signalling overhead for maintaining the reservations is

reduced. The application can be informed of the current QoS provided between the anchor point and the BAR for the traffic via the refresh messages generated for the reservation.

The anchor point maintains information about the BAR to which the mobile is currently attached in order to correctly tunnel the packets. If this information is soft-state, the refresh rate of this information and the refresh rate for the RSVP reservation can be related because it is only valid if traffic can be forwarded to the mobile. This reduction is dependent on how the reservation is repaired across the intermediate routers in the event of a network node failure. If standard RSVP is used, then the refresh rate must be set to a value that can re-install the reservation with a minimum latency after a network node failure. If an RSVP mechanism with local path repair and a more hard state operation is deployed, then the refresh rate can be the same as that used to refresh the location information at the anchor point. It is possible for a completely hard state RSVP implementation to be used, but the anchor point must explicitly remove the reservation when the location information for a mobile times out. Also, some mechanism to release the resources in the event of anchor point failure is required.

A4.5.3.3 Handover

There are two types of handover that are considered:

- inter-BAR handover: the MN changes the BAR to which it attached but maintains the same CCoA and anchor point
- inter-anchor point handover: the MN changes anchor point and is allocated a new CCoA

Within these two categories of handover the distinction between planned and unplanned handover can also be made.

Whichever type of handover occurs, a new tunnel, and therefore a new reservation, must be created between the anchor point and the BAR. This is because the packets must be classified according to the tunnel header information, which will change after every handover. The reservation for the new tunnel cannot be installed until the path update information has propagated to the anchor point, otherwise the route to the mobile is unknown. The anchor point or the old BAR is also responsible for removing the reservation associated with the old tunnel unless it is just left to timeout, which is not the most efficient use of the BAN resources.

Disruption to QoS can be minimised by introducing a context transfer phase into the handover mechanism. From the QoS perspective, the context transfer can exchange the mobile's QoS requirements between the nodes involved in the handover. For planned handover, the new BAR or anchor point can be selected based on whether they have enough resources available to support the mobile after handover. In fact, the reduction in resource availability at a BAR or anchor point, or a dramatic loss in the QoS provided to the traffic flows, could be the trigger for a handover to occur. Any modifications to the QoS required by the application flows after an inter-BAR handover can be signalled to the anchor point in the path update message generated by the BAR. This will trigger one or more reservation creations by the anchor point depending on the level of aggregation used across the BAN. During inter-anchor point handover, information about the QoS requirements of a mobile can be signalled to the new anchor point either using a specialised signalling protocol, or by including the QoS information in the path update message used to register the mobile's location with the new anchor.

The benefits of including the QoS information in the path update messages is that the reservation will not be installed until a valid route to the mobile's new location is propagated into the network, and the reservation is installed at the same at the same time as the tunnel. The triggering of the RSVP signalling by the path update messages ensures that the reservation is installed as soon as possible after the route to the mobile is known.

During inter-BAR handover, a temporary tunnel is used between the old and the new BAR to avoid packet loss after the mobile has changed location and before the path update information has propagated into the BAN. QoS provisions must be made for this temporary tunnel in order to ensure that the packets arrive at the mobile within the confines of the agreed QoS contract between the MN and the BAN. Provisioning of resources for temporary tunnels depends on whether the handover is planned or unplanned. It is unfeasible to set-up a reservation for traffic in the unplanned handover case, and a set of pre-defined DSCPs can be used. Unplanned handovers will have a dramatic effect on the QoS received by an application because there is a period of time where the location of the mobile is unknown, so packets cannot be forwarded. Planned handovers can use the same DSCPs or alternatively, a number of reservations can be signalled between BARs. There is a trade-off between the QoS received by the forwarded traffic and the signalling overhead associated with providing it. For example, the overhead associated with creating an RSVP reservation between the old and new BARs for each forwarded traffic

flow is unfeasible for such a brief period of time. Therefore, it may be desirable to aggregate the traffic flows onto one temporary reservation, but at the cost of losing the flow isolation and strict QoS guarantees.

A4.5.3.4 Session Termination

Session termination has no interaction with the micro-mobility protocol, and operates the same as in fixed networks.

A4.5.4 QoS Interaction with Tunnels

This section provides an overview of the problems associated with provisioning QoS for IP tunnels. An IP tunnel encapsulates traffic in another IP header as it passes through the tunnel. Additional headers may be inserted after the encapsulating IP header. The routers between the tunnel endpoints use the information in the outer IP header to process the packet, and require a way to determine the QoS the packet should receive. This discussion does not consider tunnels other than the IP tunnels described above.

Tunnels are employed by the BCMP to forward traffic from the ANP to the BAR and between BARs during handover. The tunnels between the BARs are only temporary.

When the packets are marked using DSCPs, the tunnel endpoints can use the information in the original header to determine suitable per-hop behaviour for the packet as it traverses the tunnel. This can be marked in the outer IP header, and used by the intermediate routers to forward the packet with the correct QoS. In the simplest case, the DSCP from the inner header is copied into the outer header. However, in order to increase flexibility, it may be desirable to allow the propagation of the DSCP and/or some of the information that it contains to the outer IP header on ingress and/or back to inner IP header on egress. Further details of how this can be achieved, and the extra complexity required in the network nodes to support it, are provided in [A4.42], [A4.43].

RSVP signalling and provisioning of reservation-based QoS over IP tunnels is more complex. RSVP packets transmitted through a tunnel are not recognised as RSVP signalling messages by the intermediate routers and pass transparently through without making the necessary QoS provisions for the traffic flows. Also, packets are classified in RSVP routers according to information both in the IP header and the transport protocol header. When packets are tunnelled, the original information by which the flows are recognised is not available.

A proposed solution to this issue [A4.4] makes the tunnel endpoints responsible for establishing a reservation between them for the traffic flows. The original RSVP message is used by the ingress tunnel endpoint to generate an RSVP message that will reserve resources across the intermediate routers to the tunnel egress point. The session description included in these messages use the tunnel header information. The original RSVP message is forwarded through the tunnel to the destination to reserve resources across the rest of the network. An additional UDP header is also inserted between the inner and outer IP headers that contains the port information required to distinguish between the traffic flows.

Reservations for the tunnels can be provided on a per-flow or an aggregate basis. If provided per flow, the signalling overhead for provisioning the QoS for the traffic flow is quite high, and the tunnel endpoints have to maintain quite a lot of state information for each flow (the state information for the original and tunnelled RSVP sessions). Alternatively, a single reservation for the tunnel can be created over which multiple flows can be aggregated. Aggregation provides a way to reduce the signalling overhead, computational overhead and memory required in routers in heavily loaded regions of the network [A4.6]. However, this benefit is offset by the loss of flow isolation which means that a flow passing through the aggregate reservation may be suffer delay from the bursts of another. The QoS provided for the flows within the aggregate can be differentiated using DSCPs if required.

There is a certain amount of complexity required at the tunnel endpoints in order for the QoS reservation for the aggregate flows to be calculated and to carry out the mapping between the incoming RSVP messages and those required for the tunnel. Support for the SESSION_ASSOC [A4.4] object is also required to associate the RSVP messages of the end-to-end and tunnel reservations.

In terms of the BCMP, the benefit of tunnels is that all complexity is pushed to the BRAIN-specific network elements, the ANPs/BMGs and the BARs. These edge devices are special purpose and will always contain per-host information. The inter-mediate nodes within the BAN can just be standard IP routers. In this scenario, QoS routing and traffic engineering can be supported using standard IP mechanisms and protocols already defined by the IETF.

A4.5.5 Combined L2/3 signalling

Problem Address:Signal Minimisation
Negotiations for network layer QoS resources are dependent on the completion of the link layer QoS negotiations, which are aided by awareness of network layer QoS requirements. It is desirable for the network layer QoS negotiations to commence as soon as the QoS provided by the link layer is known. This is not currently supported in existing link layer signalling protocols.

Network layer QoS parameters could passed to the link layer through the IP2W interface and are carried across the air interface by the link layer signalling messages. The network layer QoS parameters would be carried as transparent data, and passed back through the IP2W interface at the other side to network layer. There, they can then be used either to provision QoS resources in the network or provide feedback to the application. The signalling occurs when the allocation or re-negotiation of QoS resources allocation is required and possibly after handover.

This approach would remove the need for the MN to perform layer 4 (end-to-end) or layer 3 (network layer) QoS signalling, as the BAR could generate this on behalf of the mobile. Thus is could reduce significantly the signalling load on the MN by preventing duplication of information within different protocols and by allowing a highly optimised protocol to be used at the link layer. However, this approach breaks all the design principles – it prevents the mobile operating in true end-to-end fashion and breaks the layering principles. Therefore this approach has not been deeply considered with the project.

A4.6 Quality of Service References

[A4.1] J. Wroclawski, "The Use of RSVP with the IETF Integrated Services", IETF RFC2210, September '97.

[A4.2] J. Wroclawski, "Specification of Controlled-Load Network Element Service", IETF RFC2211, September '97.

[A4.3] S.Shenker et al, "Specification of Guaranteed Quality of Service", RFC2212, September '97.

[A4.4] Braden, R., Zhang, L., Berson, S., Herzog, S., Jamin, S., "Resource ReSerVation Protocol (RSVP) -- Version 1, Functional Specification". Internet Engineering Task Force, Request for Comments 2205, September 1997.

[A4.5] A. Terzis et al, "RSVP Operation over IP Tunnels", RFC2746, January '00.

[A4.6] F. Baker at al, "Aggregation of RSVP for IPv4 and IPv6 Reservations", Internet Draft (work in progress), draft-ietf-issll-rsvp-aggr-03.txt, Feb '01.

[A4.7] P. Eardley, "QoS Interaction and MER_TORA", IST-1999-10050/BT/WP2/PI/I/007/a1, August '00.

[A4.8] E. Gustafsson, A. Jonsson, and C. Perkins, "Mobile IP Regional Registration", Internet Draft (work in progress), draft-ietf-mobileip-reg-tunnel-02, March '00.

[A4.9] K. El. Malki , N.A. Fikouras. S.R. Cvetkovic, "Fast Handoff Method for Real-Time Traffic over Scaleable Mobile IP Networks", Internet Draft (work in progress), draft-elmalki-mobileip-fast-handoffs-01.txt, June '99.

[A4.10] C. Castelluccia, "A Hierarchical Mobile IPv6 Proposal", Technical Report No 0226 INRIA, November '98.

[A4.11] R. Ramjee, T. La Porta, S. Thuel and K. Varadhan, "IP micro-mobility support using HAWAII", Internet Draft, (work in progress), draft-ietf-mobileip-hawaii-00, June '99.

[A4.12] S. Seshan, H. Balakrishnan and R. H. Katz, "Handoffs in Cellular Wireless Networks: The Daedalus Implementation and Experience", ACM/Baltzer Journal on Wireless Networks, '95.

[A4.13] A. Mihailovic, M. Shabeer and A. H. Aghvami, "Multicast for Mobility Protocol (MMP) for emerging internet networks", Pro Proceedings of the eleventh IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2000), September 2000, London UK.

[A4.14] C. Tan, S. Pink, and K. Lye, "A Fast Handoff Scheme for Wireless Networks", In Proceedings of the 2nd ACM International Workshop on Wireless Mobile Multimedia, ACM, August '99.

[A4.15] A. O'Neill, G. Tsirtsis, and S. Corson, "Edge Mobility Architecture", Internet Draft (work in progress), draft-oneill-ema-01.txt, March '00.

[A4.16] V. Park and S. Corson, "Temporally-Ordered Routing Algorithm (TORA) Version 1 Functional Specification", Internet Draft (work in progress), draft-ietf-manet-tora-spec-02.txt, October '99.

[A4.17] K. El Malki and H. Soliman, "Hierarchical Mobile IPv4/v6 and Fast Handoffs", Internet Draft (work in progress), draft-elmalki-soliman-hmipv4v6-00.txt, March '00.

[A4.18] "QoS routing in networks with inaccurate information" Guerin and Oroa IEEE/ACM transactions on networks Vol 7 # 3 June 1999

[A4.19] Joens, Y, Goderis, D, et al., "Specification and Usage Framework", draft-manyfolks-sls-framework-00 (work in progress), October 2000

[A4.20] Bernet, Y. et al, "A Framework for Integrated Services Operation over DiffServ Networks". Request for Comments 2998, Internet Engineering Task Force, November 2000.

[A4.21] Bernet, Y., Blake, S., Grossman, D., Smith, A., "An Informal Management Model for DiffServ Routers". Internet Draft (work in progress), July 2000 (draft-ietf-diffserv-model-04.txt).

[A4.22] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., Weiss, W., "An Architecture for Differentiated Services". Internet Engineering Task Force, Request for Comments 2475, Dec. 1998.

[A4.23] Braden, R., Clark, D. Shenker, S., "Integrated Services in the Internet Architecture: an Overview ". Internet Engineering Task Force, Request for Comments 1633, June1994.

[A4.24] Huston, G. ”Next Steps for the IP QoS Architecture”. Internet Architecture Board, Request for Comments 2990, Internet Engineering Task Force, November 2000.

[A4.25] Wroclawski, J., “ The Use of RSVP with IETF Integrated Services”. Internet Engineering Task Force, Request for Comments 2210, September 1997.

[A4.26] Jacobson, V., Nichols, K., Poduri, K., “The ‘Virtual Wire’ Per-Domain Behaviour”. Internet Engineering Task Force, Internet Draft, July 2000 (drat-ietf-diffserv-pdb-vw-00.txt).

[A4.27] Berners-Lee, T., Fielding, R., Frystyk, H., "Hypertext Transfer Protocol -- HTTP/1.0". Internet Engineering Task Force (RFC) 1945, May 1996.

[A4.28] Ramakrishnan, k., Floyd, S., “A Proposal to add Explicit Congestion Notification (ECN) to IP”. Internet Engineering Task Force, Request for Comments (RFC) 2481, January 1999.

[A4.29] Yavatkar, R., Pendarakis, D., Guerin, R., “A Framework for Policy-based Admission Control”. Internet Engineering Task Force, Request for Comments 2753, January 2000.

[A4.30] Boyle, J., Cohen, R., Durham, D., Herzog, S., Rajan, D., and Sastry, A., “COPS Usage for RSVP”, Internet Engineering Task Force, Request for Comments 2749, January 2000.

[A4.31] Boyle, J., Cohen, R., Durham, D., Herzog, S., Rajan, R. and Sastry, A., "The COPS (Common Open Policy Service) Protocol". Request for Comments 2748, Internet Engineering Task Force, January 2000.

[A4.32] Yavatkar, R., et al. “SBM (Subnet Band-width Manager”. Internet Engineering Task Force, Request for Comments 2814, May 2000.

[A4.33] Conta, A., Deering, S., ”Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification”. Internet Engineering Task Force, Request for Comments (RFC) 2463, December 1998.

[A4.34] Chimento, P, Geib, R et. al., “QBone Bandwidth Broker Architecture”, http://qbone.inetrnet2.edu/bb/bboutline2.html

[A4.35] Castelluccia, C., Bellier, L., “Hierarchical Mobile IPv6”, Internet Draft (work in progress), July 2000 (draft-castelluccia-mobileip-hmipv6-00.txt).

[A4.36] Gustafsson, E., Jonsson, A., Perkins, C,. “Mobile IP Regional Registration”, Internet Draft (work in progress), July 2000 (draft-ietf-mobileip-reg-tunnel-03).

[A4.37] Malinen, J., Perkins, C., “Mobile IPv6 Regional Registrations”, Internet Draft (work in progress), July 2000 (draft-malinen-mobileip-regreg6-00.txt).

[A4.38] El Malki, K., Soliman, H., “Hierarchical Mobile IPv4/v6 and Fast Handoffs”. Internet Draft (work in progress), March 2000, (draft-elmalki-soliman-hmipv4v6-00.txt)

[A4.39] O'Neill, A., Tsirtsis, G., Corson, S., “Edge Mobility Architecture”, Internet Draft (work in progress), draft-oneill-ema-01.txt,March 2000.

[A4.40] Carter et al, “A bounded delay service for the Internet” IETF Internet Draft (expired) draft-carter-bounded-delay-00.txt, 1998

[A4.41] “QoS architectures for connectionless networks”, Fallis and Hodgekinson, IEE Colloquium on Control of Next generation Networks, London 1999

[A4.42] S. Blake et al, “An Architecture for Differentiated Services”, IETF, RFC2475, December 1998.

[A4.43] D. Black, “Differentiated Services and Tunnels”, IETF, RFC2983, October 2000.

[A4.44] B. Carpenter, “Architectural Principles of the Internet”, IETF, RFC1958, January 1996

[A4.45] Mitzel, “Overview of 2000 IAB Wireless Internetworking Workshop”, IETF, RFC3002, December 2000.

[A4.46] Montenegro et al, "Long Thin Networks", IETF, RFC2757, January 2000

[A4.47] Casner and Van jacobson, “Compressing IP/UDP/RTP headers for low speed serial links” IETF, RFC2508, 1999

[A4.48] Braden et al, “Recommendations on queue management and congestion avoidance in the internet”, IETF, RFC2309, April 1998

[A4.49] K. Nichols et al, “A two bit differentiated services architecture for the Internet”, IETF, RFC2638, July 1999

[A4.50] Silvano et al, “RSVP Proxy”, IETF Internet draft (expired) July 2000

[A4.51] Y. Bernet, “Format of the RSVP DCLASS object”, IETF, RFC2996, Novemebr 2000

[A4.52] Handley et al, “SIP:Session Initiation Protocol”, IETF, RFC2543, March 1999

A5 Enhanced Socket Interface Annex

A5.1 Introduction

The purpose of this document is to specify a QoS supporting transport service interface (The term *service interface* is being used according to [A5.8], [A5.15]). The basic idea of such a *QoS supporting transport service interface* is to use the already well-known transport primitives[53] and *enhance* it by additional primitives to give applications the facility to use QoS. Application, designed by WP1, can be implemented against this QoS supporting transport service interface. As depicted in Figure A5-1 this transport service interface is named *Enhanced Socket Interface*, abbreviated *ESI.*

The *Enhanced Socket Interface* is a generic interface, which means it is independent of any platform[54], supported QoS and any transport service provider. Since it makes sense to have also a service interface tied to the specific transport and QoS Service Provider, further *interfaces* are introduced. Representative of this kind of interface is the *Brain Service Interface* abbreviated *Brain SI.*

The Enhanced Service Layer (ESL) provides with the functionality needed by the *ESI.* It consists mainly of a QoS Mapper- and Primitive Mapper-Entity, which are mapping *ESI* functionality to the available transport, and/or QoS Service Providers. The *ESL* is located between the Application and Transport layer, the latest represented by the *BRAIN-SI.*

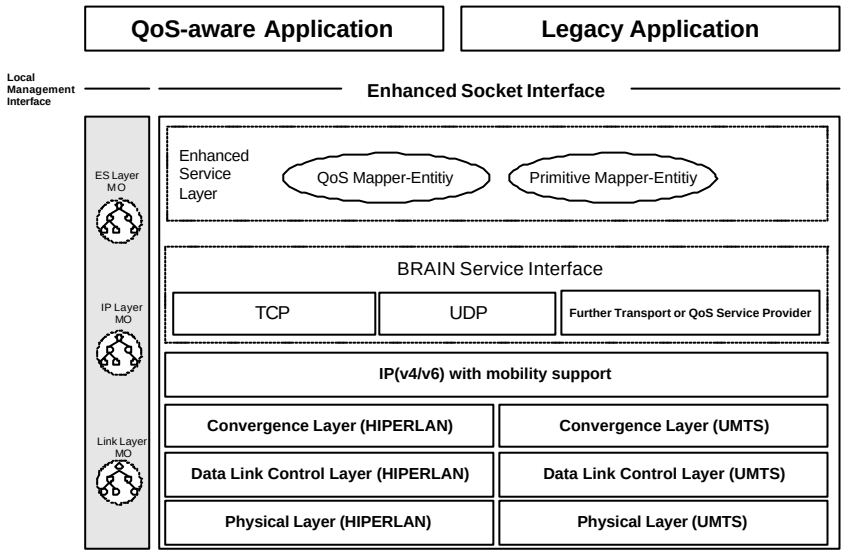


Figure A5-1: Overall Architecture

This leads to the following components, part of the overall architecture:

1. A platform, QoS and transport Service Provider independent interface, close to the application, called *Enhanced Socket Interface*
2. A Transport and QoS Service provider specific *interface* like *BRAIN-SI*
3. The *ESL* responsible for Mapping *ESI* functionality onto available Transport and QoS Service Provider (*BRAIN-SI*).
4. Local Management functionality, which can be access through the *Local Management Interface*

In the following document we want to explain each component one after the other. It is done in a top down approach, first considering the *ESI* from the *BRENTA* point of view, going to *ESI* and the *BRAIN-SI* and explaining the functionality of the *ESL.* The document concludes with an exhaustive appendix containing technical background knowledge and a glossary.

[53] like open, close, listen, connect - see for example BSD 4.3 socket interface.

[54] In contrast to Microsoft GQoS which can only be used with Microsoft's operating systems

A5.2 Quality of Service

Before going in architectural and *ESI* related details it is started with an introduction about QoS. Quality of Service in general is described as "The collective effect of service performance, which determines the degree of satisfaction of a user of the service" (see [A5.16]). In this document, it is focused on the QoS part that is determined by the network. The QoS mechanisms described provides applications with a means by which network resources—such as available bandwidth and latency performance—can be managed on both local machines and on devices throughout the network so that a predictable and guaranteed service can be achieved.

With such an all-network encompassing definition, QoS functionality may require co-operation among end nodes, switches, routers, and Wide Area Network (WAN) links through which data must pass. Without some level of co-operation among those network devices, the quality of data transmission services can break down. In other words, if each of the above network devices is left to make its own decisions about transmitting data, it will likely treat all data equally, and thus provide service on a first-come first-served basis. Although such service may be satisfactory in network devices or transmission media that are not heavily loaded, when congestion occurs, such equal treatment can mean that all data passing through the device will be delayed. With this information, we can extend the definition of QoS by adding that it allows preferential treatment for certain subsets of data as they traverse any QoS-enabled part of (or device in) the network.

With the QoS capabilities described in this document, developers do not need to consider how the various components interact to achieve QoS. The components that constitute QoS implementation are instead abstracted from the QoS application development effort, allowing a single or generic QoS interface—instead of individual interfaces—for each QoS component. This provides a generic interface for the developer, and also provides a mechanism by which new QoS components (perhaps with increased functionality) can be added, without the need to completely rewriting existing QoS applications.

To achieve manageable and predictable QoS from one end of the network to the other, the collection of components that must communicate and interact results in a fairly complex process. Since applications are the driving force for requesting QoS and applications can request QoS with different stringency, we can distinguish between two basic mechanisms how QoS may be enforced:

?? ***Hard QoS Reservation:*** Here, network resources are reserved according to an application's QoS request and subject to bandwidth management policy.

?? ***Soft QoS Reservation:*** Here, network resources might be reserved according to a previous negotiated service level agreement. How the information is forwarded is subject of a per *hop behaviour* attached to the delivered information.

To enable QoS, applications have to notify somehow their requirements towards network elements, which then give preferential treatment to classifications, based on those requirements. These types of QoS can be applied to individual *flows*.

A5.2.1 QoS Contract

QoS handling between a service user[55] and a service provider can be described by a QoS contract. The QoS contract is specified on a *per flow* basis. The service user requests certain resources for a given flow to be provided and managed by the service provider. If enough resources are available to accommodate the given request a QoS contract is established between the service user and the service provider. That is the service provider guarantees[56] to treat the flow in a way that the requirements are fulfilled.

The service user on the other hand agrees that it is not going to send more traffic over the network than was specified within the QoS contract.

If both parties behave well, the QoS contract guarantees that the service provider treats all packets for the given flow in a way that the QoS requirements of the service user are met. If the service user is misbehaving (i.e. sends more traffic than it is requested in the QoS contract), the service provider gives

[55] Definition can be found in section A5.4.

[56] How tight this guarantee is, depends on the type of service that is part of a set of parameters that comprise the contract

no guarantee at all, i.e. the service provider may eventually drop packets. It is very important that the contract is not static, that is the contract can be negotiated and monitored dynamically in order to be adapted to a change in service user requirements.

A5.2.2 The concept of a QoS Service Provider

A QoS Service Provider (QoS SP) is the QoS component that implements, maintains, and handles QoS. It is in charge of handling the QoS mapped by the QoS Mapper of the *ESL*. An application cannot access a QoS SP directly rather the *ESI* primitives are mapped to the currently used QoS SP and vice versa.

A5.3 ESI from a BRENTA point of view

Work Package one defines different applications, each using the *ESI* in different ways. Figure A5-2 depicts a possible scenario, where various types of application access the *ESI* in specific ways. This chapter should give a short insight into the *ESI's* usage from BRENTA's point of view.

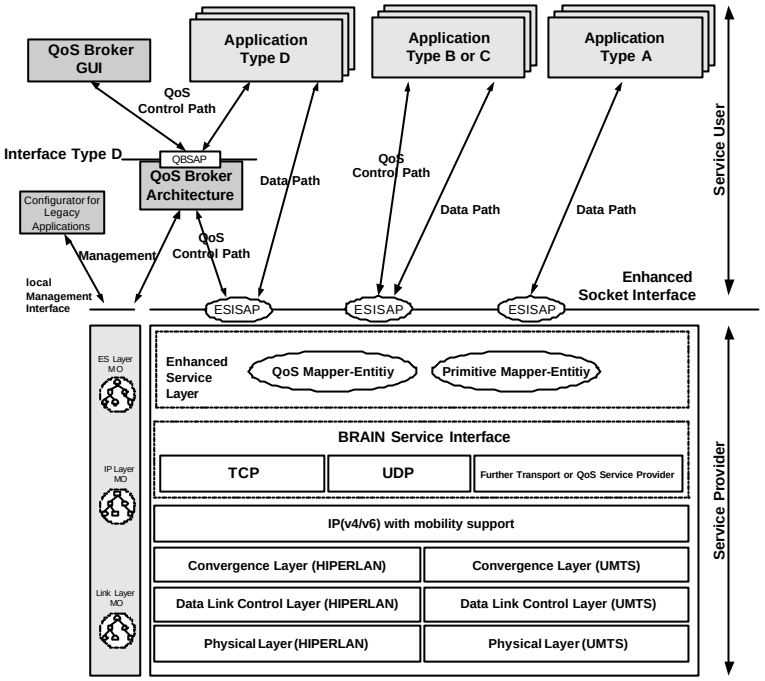


Figure A5-2: ESI used by BRENTA Applications

Legacy Application, type A

These applications are able to use only the legacy portion of the *ESI*, namely the data *send* and *receive* primitives. By using an external *Configurator* for Legacy Application, on can boost the QoS perceived by such applications.

QoS aware Application, type B and C

Applications type B and C are QoS aware applications, the difference between them being simply the fact that the latter can use multimedia components (see [A5.1]), as made available through the BRAIN Component Level API. The Multimedia components encapsulate the functionality of sending media data over network. Management related information for making QoS decision, can be retrieved through the Local Management Interface reflecting a Local Management Functionality. Work Package 1 provides component for that (see [A5.3])

QoS aware Application, type D

Application type D are QoS aware applications, envisioned to delegate to an external QoS Broker Architecture functionality the handling of QoS control related issues.

To this extent, these applications are merely concerned with data plane issues, whereas the QoS Broker Architecture takes care of QoS Control Plane issues. Thus the *ESL* shall be able to handle a given flow on behalf of two co-operating entities. Among other tasks, the QoS Broker is expected to exert some level of system administration functionality, in order to guarantee that all the applications running on the same terminal can fairly share local resources, with respect to the request QoS guarantees.

Management related information for making QoS decision, can be retrieved through the *Local Management Interface* reflecting a Local Management Functionality. Work Package 1 provides a component for that (see [A5.3])

A5.4 Modelling an Interface

Before going into detail it is introduced how a service interface can be modelled. The *ESI* is described by using the model of *Service Primitives* [A5.8], [A5.15]. A service is formally specified by a set of primitives (function calls) available. One of the ways to classify service primitives is to divide them into four classes as shown in Table A5-1. These primitives tell a service to perform some action or report on an action taken by a peer entity.

Primitive	Meaning
Request	An entity wants the service to do some work
Indication	An entity is to be informed about an event
Response	An entity wants to respond to an event
Confirm	The response to an earlier request has come back

Table A5-1: Service Primitives

Services can be either confirmed (see Figure A5-3) or unconfirmed (see Figure A5-4). In a *Confirmed Service*, there is a *request*, an *indication*, a *response* and a *confirm*. In an *Unconfirmed Service* there is just a *request* and an *indication*. A *Service.confirm*, in a *Confirmed Service*, indicates that the corresponding peer entity has received the information and sent back a confirmation.

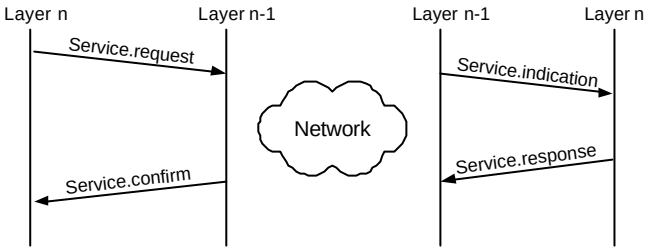


Figure A5-3: Confirmed Service

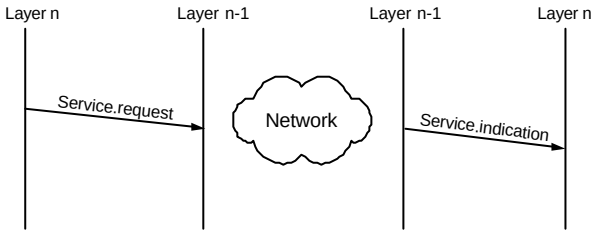


Figure A5-4: Unconfirmed Service

The *Unconfirmed Service* does not return anything, layer n passes information to the underlying layer n-1 which is responsible for sending out the information to the network. On the receiving side the information is received and passed up to layer n. The *datagram service* is an example of this kind of service which does not provide any acknowledge to the sender. The *Confirmed* and *Unconfirmed Service* are extended by primitives supporting *local error cases*. Therefore for each *Service.request* and *Service.response* primitive[57] a *ServiceErrorRequest.indication* and *ServiceErrorResponse.indication* primitive is introduced, respectively. Detailed information about the local error is passed as a primitive specific parameter. Figure A5-5 and Figure A5-6 pictures the extension.

[57] *Service.response* primitive is only available in a *Confirmed Service*.

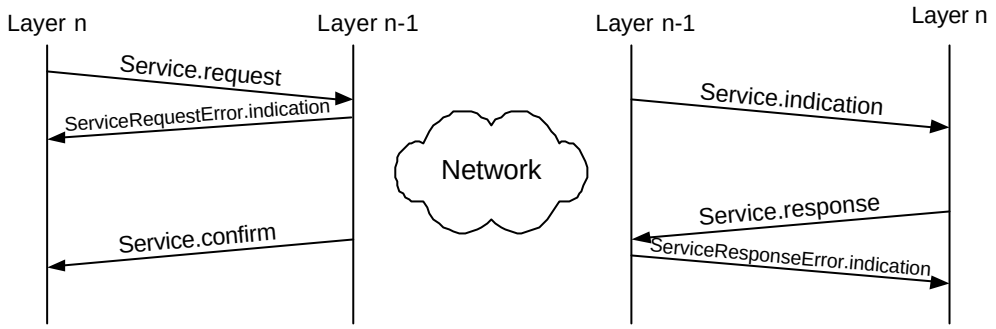


Figure A5-5: Extended Confirmed Service

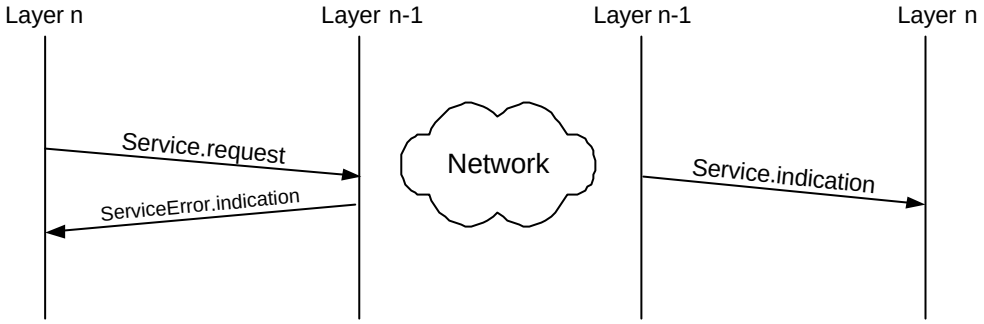


Figure A5-6: Extended Unconfirmed Service

There are application scenarios where one peer is not speaking directly with the peer across from - rather through a specific intermediate network entity like e.g. a proxy. This entity can representative confirms the request as shown in Figure A5-7. This service is named *Proxy Service* in the context of this document.

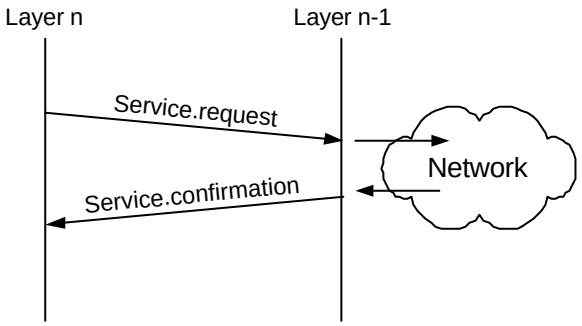


Figure A5-7: Proxy Service

The last useful case is when a network entity between the participated sides generates information. This can simply be modelled by a *Service.indication*.

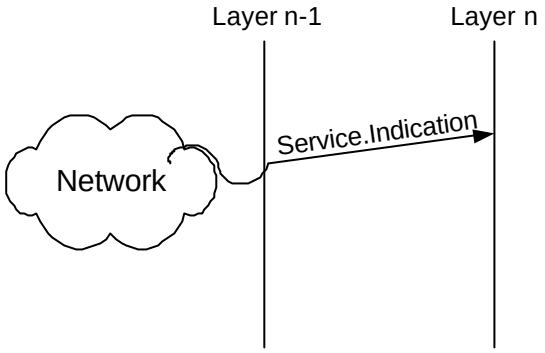


Figure A5-8: Notification Service

As described above, an interface provides not only service calls from layer n to layer n-1, but also service primitives from the lower to the upper layer, so an interface support two-way communication between a *service provider* and a *service user*.

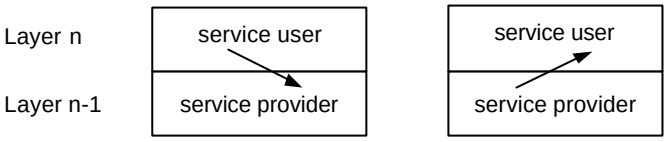


Figure A5-9: Service User / Service Provider

About Implementation

The description of the model above is rather abstract and says nothing about how it can be implemented - therefore a few words should be mentioned here. In general, the information flow from a *service user* to a *service provider* can be realised by function calls whereas the information flow from a *service provider* to a *service user* is realised by any kind of event-mechanism. Considering for example the sender side[58], in a confirmed service a *Service.confirm* follows a *Service.request* which can in a real implementation be realised as a *blocking function call*, returning the *Service.confirm's* information as the function's return value. In this case the *Service.confirm* event is caught during the execution of the function. This pattern can also be applied to the *Extended Confirmed / Unconfirmed Service's* primitive.

Another way to realise the *Service.request* and *Service.confirm* is by a *non-blocking function call*. Thereby the control flow is not blocked during execution of the *Service.request* primitive and the *Service.confirm* information is simply passed to the peer layer, which is in charge of handling the event. In the same way *Service.indication* (see Figure A5-8) can be handled. Either a *blocking call* is waiting for its *indication* or the *indication* is passed up to the peer layer.

Some words to the naming schema. In the following schema a *service* is represented by its *service primitives* (see Figure A5-3). This is only for the design of the *ESI*. In a real implementation new names might be introduced which use is more common then the names used in this document.

A5.5 Design Principles

Before starting with the detailed design of the *ESI's* primitives, the overall design principles should be worked out. This is important to get a clear understanding what the *ESI* and *ESL* are in charge of. The following items summarise the design principles applied to the *E*nhanced *S*ervice *I*nterface.

[DP 1] The *ESI* is an extension to a non-QoS aware transport service interface. It extends the ubiquitous used transport service interface by QoS primitives.

[58] It is assumed in for the sake of explanation, that the sender side accomplishes the *Service.request* and *Service.confirm* primitives.

[DP 2] The *ESI* is a generic interface, which means it is independent of any platform, supported QoS-, Network- and Transport Service Provider.

[DP 3] The *ESI* makes the development of QoS aware application possible and it supports non-QoS aware applications.

[DP 4] The *ESI* considers only *end-to-end* Quality of Service means between a Sender and a Receiver. All primitives are therefore *end-to-end* QoS related primitives.

[DP 5] The *ESI* does not introduce or enhance any existing QoS protocols - the semantic of the primitives must be realised by the available QoS Service Provider. There is no additional signalling introduced beside that of the used QoS Service Provider.

Note, a QoS negotiation protocol above the *ESI* is introduced in BRAIN Working Package 1 [A5.10].

[DP 6] It is assumed that there is a preconfigured protocol-stack with a preconfigured *ESL*[59], a connection-oriented, connection-less and at least one QoS Service Provider. The set up can be subject to the mobile user's contract with the network operator. The facility to change the default settings, especially the default used QoS- and Network Service Provider is out of scope of the *ESI* / *ESL*. (See [DP 7])

[DP 7] Local Management Issues like information about available QoS, Transport or Network Service Providers are not considered in the *ESI*. Their functionality is part of the *Local Management Functionality* and can be access through the *Local Management Interface*.

A5.6 Enhanced Service Layer

The Enhance Socket Layer abbreviated *ESL* is located between the Application and Transport layer as depicted in Figure 5-1: . It provides at least with two entities a QoS- and Primitive Mapper. Their functionality is to map the *ESI primitives* and their associated QoS Parameters to the *ESL* aware Service Providers, especially the aware QoS Service Providers. For the sake of this introduction, it is assumed that the Connection-oriented and Connection-less Service Provider are always available, whereby the availability of QoS Service Providers might change. Therefore *ESL aware* in this context means, that the *ESL* has to be informed about the currently usable QoS Service Provider. If due network provider changes or other effects a new QoS Service Provider becomes available or an in use QoS Service Provider can not longer be used, the *ESL* has to be informed about that. An open and in this document not discussed issues[60] are a) the handover of existing flows between different QoS Service Providers and b) the hand-over between different Service Interface (i.e. the BRAIN-SI and the UMTS-SI)[61].

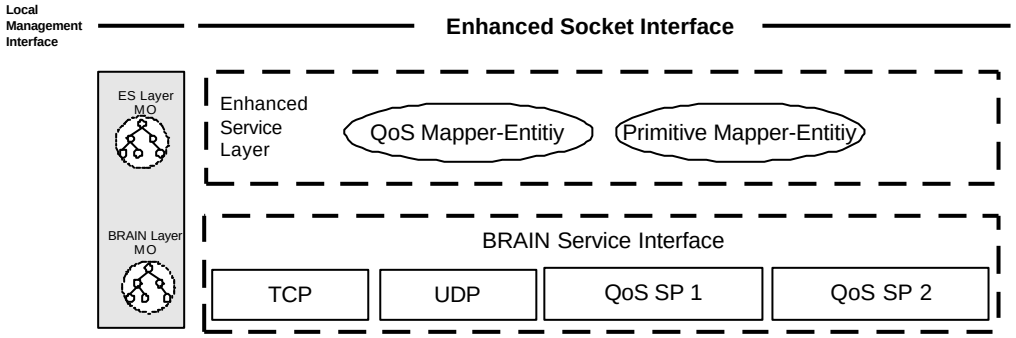


Figure A5-10: Enhanced Service Layer

[59] The *ESL* must know which QoS Service Providers are available and which appropriate QoS- and Primitive Mapper must be set up.

[60] This might be a task for the follow up project of BRAIN *MIND*.

[61] This might involve the *Local Management Interface*.

Figure A5-10 shows both entities responsible for the mapping between the ESI and the BRAIN-SI. The fact of mapping is important for the understanding of the *ESI primitives*. They are independent on the available QoS Service Provider. It is not possible for the *ESI* do say in general anything about the detail of the QoS Service Provider capabilities - more exact anything about the QoS Service Provider Protocol capabilities. It is simply not possible for the *ESI.* The *ESI* supports different kind of service (as explained in the next chapter), but how there are realised in detail is not visible for the *ESI.*

Local Management Functionality can be accessed through the *Local Management Interface*. If the *Enhanced Service Layer* makes the configuration of the QoS- or Primitive Mapper available through the *Management Functionality* it is possible for the application to influence the mapping behaviour. But this depends totally on to which extent the *ESL* gives access to their Mapper's properties. The same is valid for the *BRAIN Layer Management Functionality*. The application can access information about the current used QoS Service Provider and their capabilities through *the Local Management Interface*. But again, only these properties can be accessed and modified which are made available by the specific layer.

A5.7 Design Decisions

The last two chapters introduced the requirements to the *ESI* and its dependency on the mobile terminal usable QoS Service Provider. The *ESI* supports QoS aware application with a very generic[62] interface. Due to the independence from the used QoS Service Provider no detailed information can be offered to the upper layer.

It can be distinguished between two main characteristics of QoS Service Provider, namely the capability of supporting explicit end-to-end signalling, supported in QoS Service Provider like RSVP (see [A5.6]) and YESSIR (see [A5.11]) and QoS Service Provider which do not support any explicit end-to-end signalling like DiffServ (see [A5.14]). One should be careful to associate explicit end-to-end signalling with reservation of resources between sender and receiver (see [A5.13]). It might be if all router between the participant supports reservation but it is hard to assume that. Explicit end-to-end signalling enables the design of a confirmed service, given the upper layer information about the request QoS. Non-explicit end-to-end signalling might not support any information to the upper layer, it might be that the upper layer does not know if the requested QoS is realised in any form.

The following main characteristics serve as the basis for the *ESI*:

[DD 1] A confirmed service, supporting the upper layer with information about whether the requested QoS can be supported or not. Therefore an appropriate explicit end-to-end signalling protocol must be available.

[DD 2] An unconfirmed service, which does not support the upper layer with information about whether the requested QoS can be supported or not. Information is sent immediately without considering explicit end-to-end signalling - means no overhead in explicitly establishing a QoS aware flow.

[DD 3] Notification Service, indicating the violation of a QoS aware flow.

[62] The term generic is used in many different ways. Here it should point out that the *ESI* is platform, QoS and Transport Service Provider independent.

A5.8 Enhanced Socket Interface

This section covers the definition of the *ESI's* primitives, based on the model defined in section A5.4. The basic design principles used are introduced in the previous section A5.5 and A5.7, so starting now with the detail of the desired functionality of the *ESI.* The main issue of the *ESL* is to offer services - providing QoS for a given established flow[63]. The service offered to the upper layer of the *ESL* can be either confirmed or unconfirmed. The following sections describe the *ESL's* services offered through the *ESI* to upper layers.

It should be noted, as discussed in detail in the last chapter, that the *ESI* is independent from the used QoS Service Provider. The exact semantic of the *ESI's* is subject to the *Primitive* - and *QoS Mapper* introduced later in this document. These Mappers define how *ESI primitives* and their *Parameters* are mapped to the available QoS Service Provider's primitives. The description found in the following sections are very general, entail no assumption to the QoS Service Provider.

Some words should be said about the usage of a confirmed service. Assume a service user is requesting a specific service with *Service.request* primitive. This request is acknowledged with a *Service.confirm* primitive. It is worth to note that for the service user it might be not possible to find out which network entity has triggered the *Service.response* primitive. In the case of using a QoS Service Provider aware Proxy any network element between the actual *Sender* and *Receiver* can act as a proxy and triggering the associated *Service.response* primitive. The real behaviour depends inherent on the thereby used QoS Service Provider[64].

A5.8.1 SetQoS, a confirmed service of the ESL

The confirmed service acknowledges whether the requested QoS for a specific flow can be granted or not. Therefore reservation in all participating nodes[65] between the sender and receiver (inclusive) has to be accomplished. The confirmation will be a positive acknowledge if the end-to-end QoS can be granted or a negative one if not. Due the above defined design principles [DP 5] this service can only be offered if the protocol stack provides with an appropriate end-to-end signalling capable QoS Service Provider, if not this kind of service can not be supported to the upper layer. To model the *SetQoS* service we use the already introduced confirmed service (see Figure A5-3).

There are different models how QoS is established between a Sender and a Receiver. It has to be distinguished between Sender-initiated reservation, done in Yessir (see [A5.11]), and Receiver initiated reservation done in RSVP (see [A5.6]). This fact is held on in several primitives for example *SetQoS.request* and *SetQoS.response*. Assume for example the availability of an RSVP aware QoS Service Provider[66] in the mobile terminal. During establishing a QoS aware flow the Receiver of the flow is informed by a *PATH* message containing information about the path[67]'s characteristics and the flow's QoS requirement.

Usage: Extended Confirmed Service model, see Figure A5-5

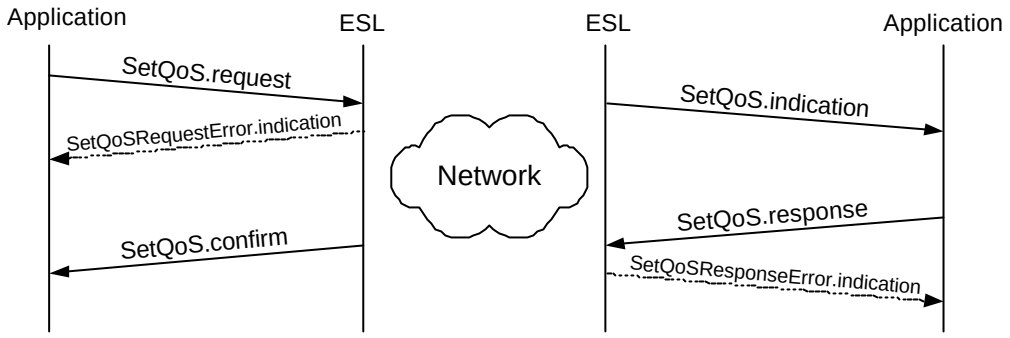
[63] A definition of the term flow used in this document can be found in the Glossary.

[64] It might possible e.g. in RSVP that the creator of the RESV message can be identified by its IP address. Based on that information the sender can find out if it is communicating with the corresponding host or a proxy.

[65] Means all nodes participating in the used explicit end-to-end signalling protocol. Might be the BAR and BMG of the BRAIN Access Network. (TODO:OS insert reference to the QoS Group's document)

[66] According to RSVP Version 1

[67] The route between the sender and the receiver



SetQoS.request primitive	service user -> service provider
associated information	flow, to be associated with QoS QoS, QoS parameter (see **QoS Parameters**)
local error cases	Handled through the *SetQoSRequestError.indication* primitive indicating: If an explicit end-to-end signalling capable QoS Service Provider is available in the current configuration, there might be the following local error cases: If it is not allowed - in the role of a receiver - to use the *SetQoS.request* primitive for the given flow, **SETQOS_REQUEST_NOT_ALLOWED_FOR_RECEIVER** is passed as parameter and no further processing is done. If it is not allowed – in the role of a sender – to use the *SetQoS.request* primitive for the given flow, **SETQOS_REQUEST_NOT_ALLOWED_FOR_SENDER** is passed as parameter and no further processing is done. If there is no explicit end-to-end signalling capable QoS Service Provider available in the current configuration **SETQOS_NO_E2E_SIGNALLING** is passed as parameter and no further processing is done.
description	Requests QoS to be associated with the given flow. It is up to the selected QoS Mapper and the QoS SP to interpret the abstract QoS parameters (see **QoS Parameters**) and to invoke QoS treatment for the given flow according to its internal capabilities.
SetQoS.indication primitive	service provider -> service user
associated information	flow, to be associated with QoS QoS, QoS parameter (see section **QoS Parameters**)
description	The service primitive indicates that a corresponding host wants to send traffic with a certain QoS to the current host. The current host is therefore a receiver of a QoS enabled new flow yet to be established. The requested QoS, additional information e.g. about the path's characteristics if available and the associated flow is passed to the upper layer.
SetQoS.response primitive	service user-> service provider
associated information	flow, to be associated with QoS QoS, QoS parameter (see section **QoS Parameters**)

local error cases	Handled through the *SetQoSResponseError.indication* primitive indicating: If an explicit end-to-end signalling capable QoS Service Provider is available in the current configuration, there might be the following local error cases: If it is not allowed - in the role as a receiver - to use the *SetQoS.response* primitive for the given flow, **SETQOS_RESPONSE_NOT_ALLOWED_FOR_RECEIVER** is passed as parameter and no further processing is done. If it is not allowed – in the role as a sender – to use the *SetQoS.response* primitive for the given flow, **SETQOS_RESPONSE_NOT_ALLOWED_FOR_SENDER** is passed as parameter and no further processing is done. If there is no explicit end-to-end signalling capable QoS Service Provider available in the current configuration **SETQOS_NO_E2E_SIGNALLING** is passed as parameter and no further processing is done.
description	The corresponding host is informed about the current host's requested QoS via *SetQoS.response*. The responded QoS might be calculated under the aspect of a) the QoS received with the associated indication and b) additional information e.g. the path's characteristics if available. Note that the QoS can differ from the QoS passed with the indication primitive.

SetQoS.confirm primitive	service provider -> service user
associated information	flow, to be associated with QoS QoS, QoS parameter (see **QoS Parameters**)
description	This service primitive indicates the network accepted and receiver determined QoS. Note this can be different from the QoS specification passed with the *SetQoS.request* primitive.

A5.8.2 SetQoSViolation notification

As mentioned above the *SetQoS* service reflects an explicit end-to-end signalling. During establishing a QoS aware flow an error in any network entity can occur due to admission- or policy control or simply due the fact that the network entity cannot grant the requested QoS[68]. To model violation during establishing[69] of a QoS aware flow the Notification model is applied. It is worth to remember [DP 5] , *SetQoSViolation.indication* can only be supported if the underlying QoS Service Provider supports the handling of error messages, generated in network entities.

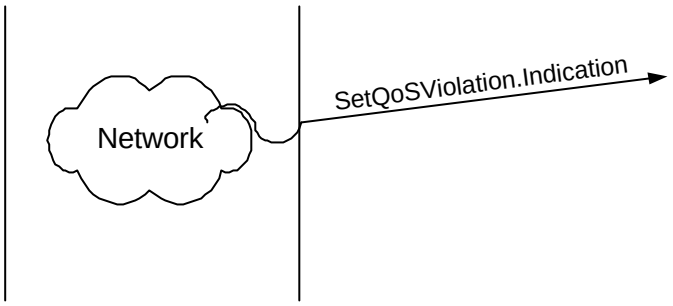
As with the usage of the *SetQoS* service it has to be considered what kind of QoS Service Provider is used. Dependent on the QoS Service Provider capabilities and how the used protocol handles violation during establishing of a QoS aware flow, the *ESI* can map the events to the upper layer.

It should be noted that it is QoS Service Provider Protocol specific who the error message receives, either the sender or the receiver. Only the entity receiving error message can map them to the upper layer.

Usage: Notification Service model, see Figure A5-8

[68] Change of the QoS can result might result in a QoS violation, but this depends on the functionality of the mobility management.

[69] QoSViolation.indication primitive is used to indicate QoS violation during the duration of the flow.



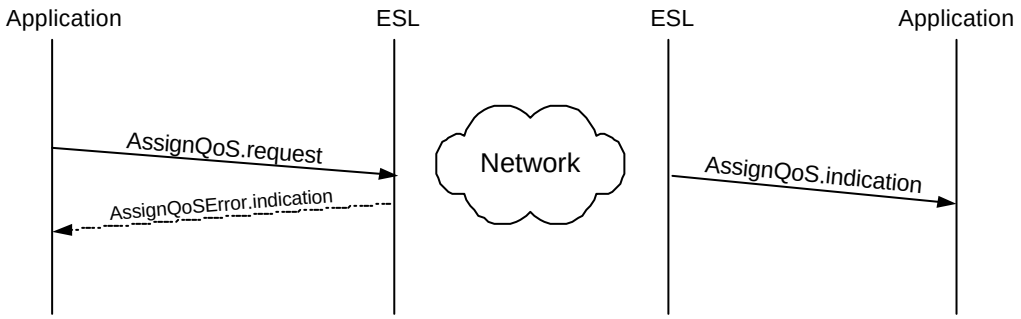
SetQoSViolation.indication primitive	service provider -> service user
associated information	flow, to be associated with QoS violation, type of violation, dependent on the used QoS Service Provider
description	Indicates that a network entity between the sender and receiver (inclusive) triggers an error during establishing a connection. The violation parameter passed with the primitive indicates the reason of the violation. The QoS Service Provider has to at least provide information about the current state of the QoS aware flow like *QoS aware flow established* or *QoS aware flow not established*.

A5.8.3 AssignQoS, an unconfirmed of the ESL

AssignQoS offers the service of delivering packets, based on specific *QoS Parameter*. There is no explicit end-to-end signalling accomplished meaning that there is no overhead in setting up a QoS aware flow and doing the reservation. The *ESL's* Mappers are in charge of mapping the requested QoS to an available QoS SP, if possible. If no not 'explicit end-to-end signalling QoS SP' is available this service cannot be supported

Note that in both services, *SetQoS* and *AssignQoS*, the same *QoS Parameters* are passed as arguments to the *ESL*. The *ESL* is in charge of mapping/filtering the QoS Parameter's information to the used QoS Service Provider.

Usage: Extended Unconfirmed Service model, see Figure A5-6



AssignQoS.request primitive	service user -> service provider

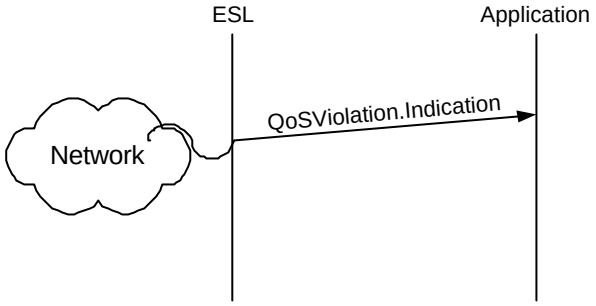
associated information	flow, to be associated with QoS QoS Parameter (see **QoS Parameters**)
local error cases	Handled through the *AssignQoSError.indication* primitive indicating: If no non-explicit end-to-end signalling QoS Service Provider is available **USEQOS_NO_NONE2E_SIGNALING** is passed as parameter and no further processing is done.
description	Requested QoS to be associated with the given flow. It is up to the selected QoS Mapper and the available non-explicit end-to-end QoS SP to interpret the *QoS Parameter* provided and to invoke QoS treatment for the given flow according to its internal capabilities. To change the QoS for a specific flow the primitive can be called repeatedly with the *new request* QoS Parameters. This means that all packets of the specific *flow* are associated with the specific *QoS* (passed as QoS Parameter). Example: *AssignQoS(flowA, QoSA),* resulting that all new generated packets for *flowA* are treated with *QoSA* *AssignQoS(flowA, QoSB),* resulting that all new generated packets of *flowA* are treated with *QoSB*
AssignQoS.indication primitive	service provider -> service user
associated information	flow, to be associated with QoS QoS Parameter (see **QoS Parameters**)
description	This service primitive indicates that a corresponding host is sending traffic with a certain QoS to the current host. The current host is therefore a receiver of a QoS enabled flow. According to [DP 5] this primitive is generated implicit during the receiving of the first QoS-marked packet of the specific. Note due the facility that the receiving side does not support this kind of QoS Service Provider or the used protocol it might be possible that this primitive is not triggered if no QoS Service Provider feels responsible for that.

A5.8.4 QoSViolation Notification

After establishing a QoS-aware flow there might be changes of the QoS between the sender and the receiver (inclusive) due the fact of interference. To inform the participating peers that the established QoS can not longer be provided the *QoSViolation.indication* primitive is triggered. This primitive is called for all QoS violations not related to establishing a QoS aware flow (see SetQoSViolation) or changing a QoS aware flow (see ChangeQoSViolation). Dependent on the QoS Service Provider capabilities and how the used protocol handles violation during sending data the *ESI* can map the event to the upper layer.

It should be noted that it is QoS Service Provider Protocol specific a) who the receiver of a *QoSViolation.indication* is, either the sender or the receiver and b) the reason for generating a QoS violation signal. Only the entities receiving a *QoSViolation.indication* can map them to the upper layer.

Usage: Notification Service model, see Figure A5-8



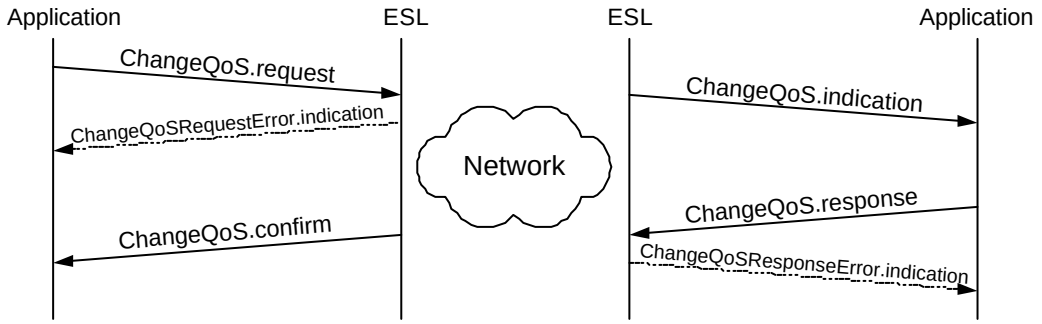
QoSViolation.indication primitive	service provider -> service user
associated information	flow, QoS aware flow violation, type of violation, dependent on the available QoS SP
description	Indicates that a network entity between the sender and receiver (inclusive) triggers an error after a QoS aware flow has been established. The violation parameter has to provide information about the state of the QoS aware flow - if it the QoS for the specific flow can be provided or not. Note that this primitive is only triggered on behalf of a specific error message of the used QoS Service Provider. The exact semantic depends on the used QoS Service Provider, QoS and Primitive Mapper.

A5.8.5 ChangeQoS, a confirmed service of the ES

After establishing a QoS aware flow with the *SetQoS* service the QoS requirements may change to alternate requirements of the receiver or sender. This leads to a change of the reserved resource between the sender and receiver (inclusive). This service can only be applied to already established QoS aware flows, set up with *SetQoS*. It is up to the currently available QoS Service Provider how the change from the old, already guaranteed QoS, to the new required QoS can be accomplished. This service can only be offered if an appropriate explicit end-to-end signalling QoS Service Provider is available.

Note, that the *ChangeQoS* service cannot be applied to QoS aware flows established with the *AssignQoS* service. As mention in the introduction of the *AssignQoS* service, a flow's QoS characteristics established with *AssignQoS* can simply be changed by a repeated call of the *AssignQoS.request* primitive with the new QoS Parameters.

Usage: Extended Confirmed Service model, see Figure A5-5



ChangeQoS.request primitive	service user -> service provider
associated information	flow, QoS aware flow QoS, new requested QoS (see QoS Parameters)
local error cases	Handled through the *ChangeQoSRequestError.indication* primitive indicating: If an end-to-end signalling capable QoS Service Provider is available in the default configuration, there might be the following local error cases: If it is not allowed - in the role as a receiver - to use the *ChangeQoS.request* primitive for the given flow, **CHANGEQOS_REQUEST_NOT_ALLOWED_FOR_RECEIVER** is passed as parameter and no further processing is done. If it is not allowed – in the role as a sender – to use the *ChangeQoS.request* primitive for the given flow, **CHANGEQOS_REQUEST_NOT_ALLOWED_FOR_SENDER** is passed as parameter and no further processing is done. If the flow is not already associated with QoS by using the *SetQoS service*, **CHANGEQOS_REQUEST_NOT_ALLOWED** is passed as parameter and no further processing is done. If there is no explicit end-to-end signalling capable QoS Service Provider available in the current configuration **CHANGEQOS_NO_E2E_SIGNALLING** is passed as parameter and no further processing is done.
description	The QoS requirement associated with the current flow has to be changed due to QoS requirements on the receiver or sender side. It is up to the selected QoS-, Primitive Mapper and QoS SP to interpret the abstract QoS parameters and to change the QoS of the current flow according the new QoS requirement.
ChangeQoS.indication primitive	service provider -> service user
associated information	flow, QoS aware flow QoS, QoS parameter (see section QoS Parameters)
description	The service primitive indicates that a corresponding host wants to change the QoS for the specific flow. The requested QoS, additional information e.g. about the path's characteristics if available and the associated flow is passed to the upper layer.
ChangeQoS.response primitive	service user-> service provider
associated information	flow, QoS aware flow QoS, QoS parameter (see section QoS Parameters)

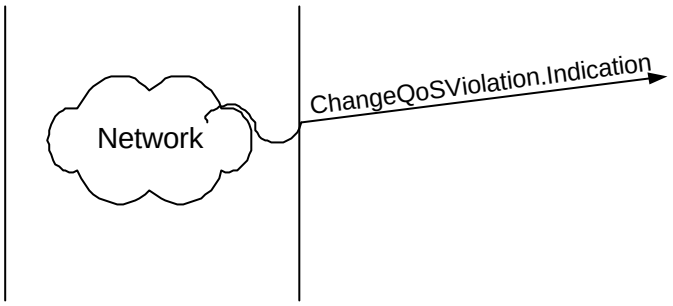
local error cases	Handled through the *ChangeQoSResponseError.indication* primitive indicating: If an end-to-end signalling capable QoS Service Provider is available in the default configuration, there might be the following local error cases: If it is not allowed - in the role as a receiver - to use the *ChangeQoS.response* primitive for the given flow, **CHANGEQOS_RES_NOT_ALLOWED_FOR_RECEIVER** is passed as parameter and no further processing is done. If it is not allowed – in the role as a sender – to use the *ChangeQoS.response* primitive for the given flow, **CHANGEQOS_REQUEST_NOT_ALLOWED_FOR_SENDER** is passed as parameter and no further processing is done. If the flow is now already associated with QoS by using the *SetQoS* service, **CHANGEQOS_REQUEST_NOT_ALLOWED** is passed as parameter and no further processing is done. If there is no explicit end-to-end signalling capable QoS Service Provider available in the default configuration **CHANGEQOS_NO_E2E_SIGNALLING** is passed as parameter and no further processing is done.
Description	The corresponding host is informed about the current host's new requested QoS via ChangeQoS.response. The responded QoS might be calculated under the aspect of a) the QoS received with the associated indication or b) additional information e.g. the path's characteristics if available. Note, that the QoS Parameter passed with this primitive might be different from the QoS Parameters passed with the *ChangeQoS.indication* primitive. The QoS might be changed by the Application.
ChangeQoS.confirm primitive	service provider -> service user
associated information	flow, QoS aware flow QoS, QoS parameter (see QoS Parameters)
Description	This service primitive indicates the network accepted and receiver determined QoS. Note this can be different from the QoS specification passed with the *ChangeQoS.request* primitive.

A5.8.6 ChangeQoSViolation Notification

As mentioned above the *ChangeQoS* service reflects an explicit end-to-end signalling. During changing a QoS aware connection an error in any network entity can occur due to admission- or policy control or simply due the fact that the network entity cannot grant the new required QoS. To model violation during changing a QoS aware flow the Notification model is applied. It is worth to remember [DP 5] , *ChangeQoSViolation.indication* can only be supported if the underlying QoS Service Provider offers error messages.

Dependent on the QoS Service Provider capabilities and how the used protocol handles violation during the change of a QoS aware flow, the *ESI* can map the event to the upper layer.

Usage: Notification Service model, see Figure A5-8

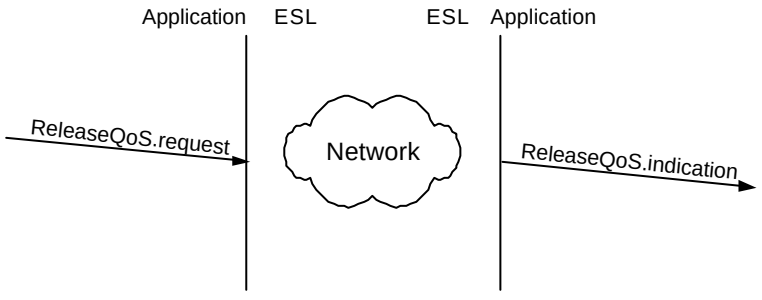


ChangeQoSViolation.indication primitive	service provider -> service user
associated information	flow, QoS aware flow violation, type of violation, dependent on the used QoS SP
description	Indicates that a network entity between the sender and receiver (inclusive) triggers an error during changing the QoS of the specific QoS aware flow. If during the change of a QoS aware flow a *ChangeQoSViolation.indication* primitive is triggered, it has to inform the receiving side (either sender or receiver) how the actual state of the QoS aware flow is. This is again inherent dependent on the capabilities of the used QoS Service Provider's Protocol.

A5.8.7 ReleaseQoS

ReleaseQoS is used for both kinds of flows, either associated with an end-to-end signalling protocol or not. If a flow is associated with an end-to-end signalling protocol, *ReleaseQoS* can be used as a unconfirmed service mapped to the QoS Service Provider *tear down* specific primitives. If a non end-to-end signalling QoS SP is used the behaviour of the ReleaseQoS service is dependent on the QoS Service Provider's nature to release the requested QoS. Again the exact meaning of the primitives depends inherent on the available QoS Service Provider and the mapping done by the QoS- and Primitive Mapper.

Usage: Unconfirmed Service for end-to-end signalling QoS SP, see Figure A5-4



ReleaseQoS.request primitive	service user -> service provider
Associated information	flow, QoS aware flow

Description	This primitive can be called with the semantic of an unconfirmed service primitive if the flow is associated with an end-to-end signalling QoS Service Provider for freeing allocated resources between the sender and receiver. If the flow is not associated with an end-to-end signalling QoS Service Provider, specific primitives of the bounded Service Providers are called. In this case *ReleaseQoS* is not an unconfirmed service primitive and no *ReleaseQoS.indication* is triggered on the peer entity.
Release.indication primitive	service provider -> service user
Associated information	flow, to be associated with QoS
Description	This service primitive indicates that a corresponding host is triggering the deallocation of QoS for the specific flow. The primitive can only be supported if the flow is associated with an end-to-end signalling QoS SP supporting a mechanism for tearing down a QoS associated flow.

A5.8.8 Summary

The following table summarises the above-defined primitives. Direction indicates the direction of the call (i.e. from service user to service provider or vice versa).

Method/Event	Meaning	Direction
Service Primitives		
SetQoS, confirmed service		
SetQoS.request	Request QoS to be associated with a given flow.	user? provider
SetQoS.indication	Indicates a new QoS connection.	provider? user
SetQoS.response	Response with probably modified QoS	user? provider
SetQoS.confirm	Confirmation of the request QoS flow. Received QoS can differ from the requested one.	provider? user
SetQoSRequestError.indication	Indicates a local error due to SetQoS.request primitive call.	provider? user
SetQoSResponseError.indication	Indicates a local error due to SetQoS.response primitive call.	provider? user
SetQoSViolation, notification		
SetQoSViolation.indication	Indicates a violation during establishing a QoS aware flow with *SetQoS*.	provider? user
AssignQoS, unconfirmed service		
AssignQoS.request	Assigns a QoS Parameter to a given flow	user? provider
AssignQoS.indication	Indicates a new QoS established flow	provider? user
AssignQoSError.indication	Indicates a local error due to AssignQoS.request primitive call.	provider? user
ChangeQoS, confirmed service		
ChangeQoS.request	Request the change of the QoS for an already QoS aware flow	user? provider
ChangeQoS.indication	Indicates that the QoS for the specific flow has to be changed to new QoS.	provider? user
ChangeQoS.response	Response with probably modified QoS	user? provider
ChangeQoS.confirm	Confirmation of the requested change for an already established QoS flow.	provider? user
ChangeQoSRequestError.indication	Indicates a local error due to ChangeQoS.request primitive call	provider? user

ChangeQoSResponseError.indication	Indicates a local error due to ChangeQoS.response primitive call	provider? user
ChangeQoSViolation, notification		
ChangeQoSViolation.indication	Indicates a violation during the change of an already, via SetQoS established, QoS aware flow.	provider? user
QoSViolation, notification		
QoSViolation.indication	Indicates a violation for an already QoS aware flow.	provider? user
ReleaseQoS, [opt] unconfirmed service		
ReleaseQoS.request	Request the release of resources associated with a given flow	user? provider
ReleaseQoS.indication	Indicates the release of resources by the corresponding peer	provider? user

A5.8.9 Usage of the ESI primitives

This section describes the necessary steps to invoke the *ESL's* QoS mechanism using the aforementioned primitives. For the non-QoS related primitives the Berkley Socket API primitives are used. This is not a requirement, but due their ubiquitous utilisation they are used in these examples. At first it is started with comprehensive examples, including all necessary steps to set up a QoS aware flow and transmit data with it. After that, special examples are explained in more detail, thereby the general set-up and shutdown procedures are not taken into account. The BRENTA architecture supports different kind of applications, (see sections A5.4) thereby for the sake of the example only the Application Type D is considered here. Note that the QoS Broker mention below is part of the BRENTA middleware. Detailed information can be found in [A5.3].

In all examples it is assumed that the mobile terminal is equipped with a) an explicit end-to-end signalling and b) with a non-explicit end-to-end signalling capable QoS Service Provider, RSVP and DiffServ respectively. This assumption is only for the sake of these examples - future mobile terminal might be equipped with many different QoS Service Provider providing different kind of QoS mechanisms. But to have a starting point - or starting QoSSPs - a RSVP and a DiffServ capable QoSSPs are assumed.

There is no assumption at all, that the whole *network* is *RSVP* aware. Figure A5-11 shows the example's network topology where RSVP is used in the customer network. It is assumed that there is a mapping from RSVP Service Types to DiffServ Service Levels. It is not the task of this document to specify the behaviour of the BRAIN Access Network capabilities in more detail see [A5.4].

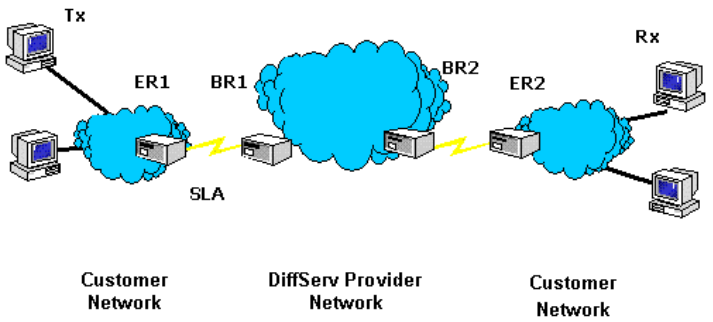


Figure A5-11: Example's Network Topology

Some words to the figures used in this chapter. As in the previous chapter *time sequence* diagrams are used to denote the relationship between the primitives that form a service and the order in which they occur. The different layer are emphasised with thick lines - thin line do not represent any layers, they are simply used for the sake of the example to give an comprehensive overview how everything plays together. The following examples should demonstrate how the most important *ESI primitives* are used for the time being.

A5.8.9.1 Usage of SetQoS

At first the usage of the *SetQoS service primitives* is explained by the means of an example describing a mobile user interested in showing a video on demand. It is expected that the transmission be not interfered - a case shown in a later example. The usage of the *ESI* primitives is presented with respect to the functionality of the application/middleware - as identified in [A5.3], to get a better understanding how everything might play together.

As depicted in Figure A5-12 the receiver (Player) and sender (Video Server) negotiate the terms of the QoS aware flow, yet to be established. This comprises e.g. QoS negotiation and exchange of information like port addresses and resource description of the movie to be played. Detailed information can be read in [A5.3]. After negotiation and exchanging set-up information both sides can logically establish a connection[70]. After establishing the connection the *Start Transmission* phase is accomplished, which includes in some way the establishment of a QoS aware flow. If the QoS aware flow is established the sender can start with the transmission of the movie till the receiver or sender starts the *End Transmission* phase which shutdowns the QoS aware flow and tears down the resources used by the specific flow. Note that the *QoSRelease* service is an unconfirmed service. It is triggered independent from the corresponding side. It might be that the sender triggers the *QoSRelease* service on behalf of for example a *RTSP Tear down Message* sent by the receiver.

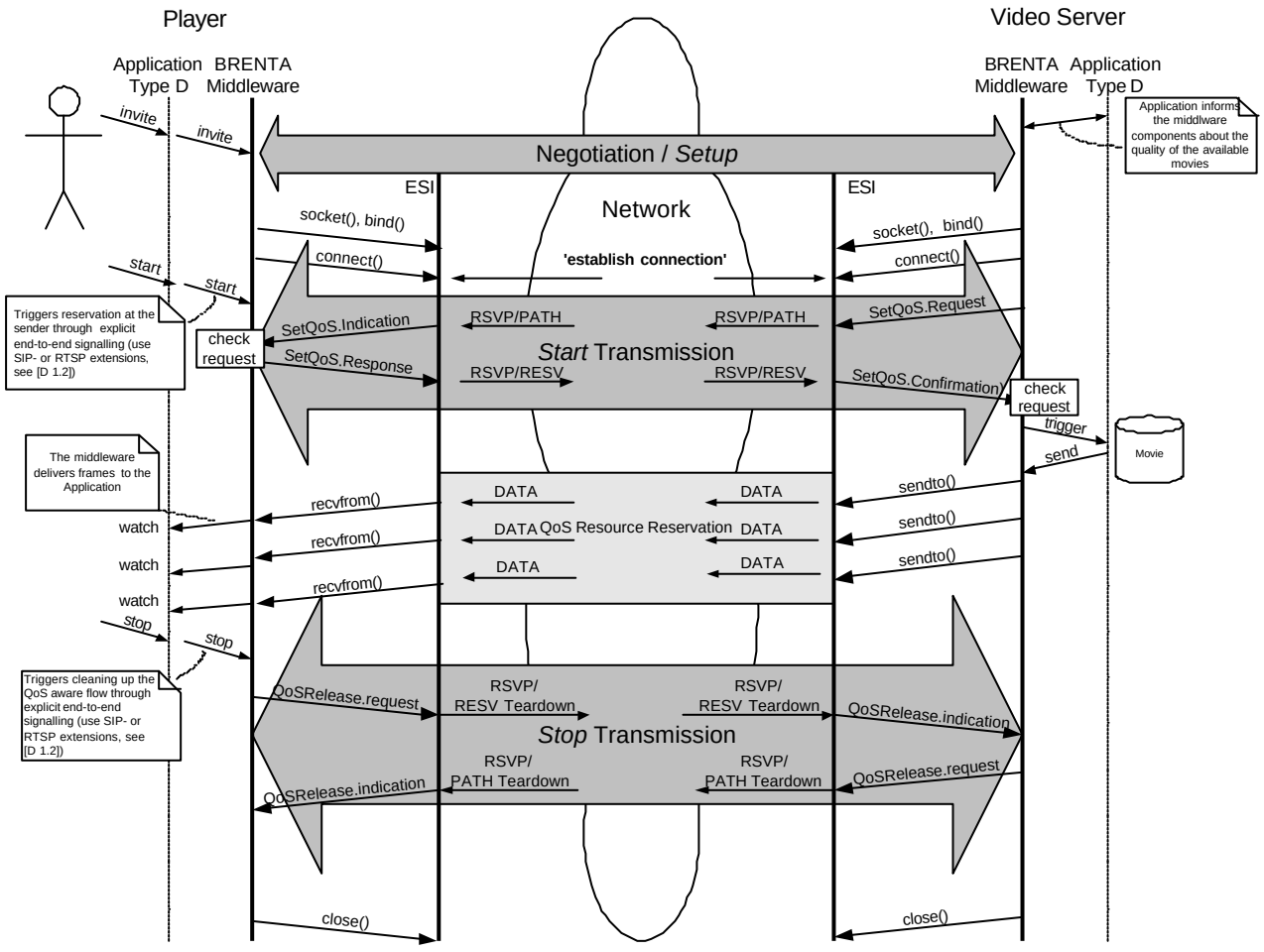


Figure A5-12 Usage of SetQoS Service primitives

In this example the QoS Broker is in charge of managing all QoS related issues[71]. The application uses a high-level interface called Type D interface for managing end-to-end QoS negotiation. Start/stop commands, which are application specific, are intercepted at this interface since they trigger resource reservation and therefore co-ordination (via end-to-end signalling) among peers, for details see [A5.3].

[70] The connection can be either Connection-oriented or Connection-less - not the subject of this example.

[71] The exact functionality of the QoS Broker can be found in [A5.3].

To come to the usage of the *ESI* primitives. They are used during the *Start* and *Stop Transmission* phases. In the *Start Transmission* phase the already established connection is associated with QoS by the means of using the *SetQoS Service*. A *SetQoS.request* primitive is called on the sender's side resulting in a *SetQoS.indication* on the receiver's side. After checking the request a *SetQoS.response* is triggered resulting in a *SetQoS.confirm* primitive call on the sender's side. It is assumed - in this example - that the used explicit end-to-end QoS Service Provider is Receiver-initiated [RSVP] not Sender-initiated like [A5.11]. For the *Stop Transmission* phase the *Unconfirmed Service QoSRelease* is used which shutdowns the connection and tear down the resources used between the sender and the receiver.

A5.8.9.2 Usage of AssignQoS

The last example demonstrates the usage of the *SetQoS Confirmed Service* in the context of watching a video on demand. Now the usage of the *Unconfirmed Service AssignQoS* is shown. The *AssignQoS Service* can be used if no explicit end-to-end signalling QoS Service Provider is available or if a sender wants to delivery a message without using the overhead of an explicit end-to-end signalling. For e.g. in the case that a mobile user wants to send a message instantly without any delay due to a) signalling exchange during establishing the QoS aware flow or b) explicit resource reservation done beside the signalling. This might be useful if the amount of information to be delivered is not so 'large'.

The *AssignQoS Service* is described here in the context of sending a multimedia message[72]. As depicted in Figure A5-13 the QoS for the specific flow can be negotiated between the sender and receiver[73] (to know which QoS Parameters to use). The important things happen in the *Start Transmission* phase where in the sender the *AssignQoS.request* primitive is called for an already logically established connection[74]. Due to the fact that there is no way to signal the receiver side about the requested QoS[75] no *Service.indication* primitive is called on the receiver side. There is the possibility, but that depends inherent on the capabilities of the used QoS Service Provider on the receiver side, that an *AssignQoS.indication* is generated due the receiving of the first packet of the QoS aware flow. For example, if the concept of marking is used, the receiving side can detect that a packet is associated with a QoS aware flow and the *AssignQoS.indication* primitive call is called.

During the *Stop Transmission* phase the *QoSRelease.indication* primitives are called on both side. Since no explicit end-to-end signalling is available these calls simply inform the lower layer that the specific QoS aware flow is now longer used. There is no stop transmission phase when sending datagrams. The only way to detect is when the socket is destroyed

[72] Message containing text, and at least either video or audio content.

[73] It is currently under discussion (WP1) if this is done.

[74] *Logically established* means that the connection has to be specified. '5-tuple (Local IP Address, Local Port, Protocol-ID, Remote IP Address, Remote Port)' (See [A5.9]). The connection can be either connection-less or connection-oriented.)

[75] This is an assumption in this case - no explicit end-to-end signalling.

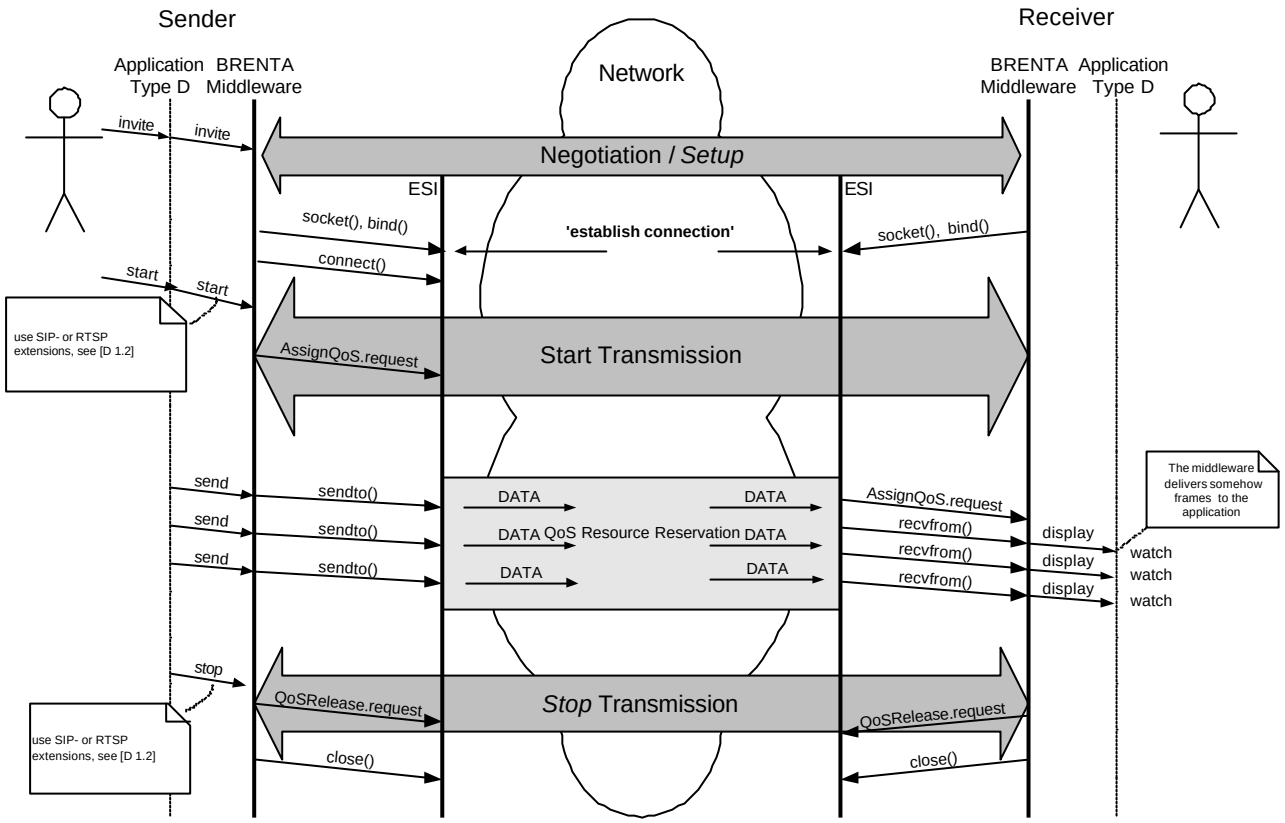


Figure A5-13: Usage of AssignQoS Service primitives

A5.8.9.3 Usage QoS Violation

The last example considers the case that a QoS aware flow, established with the *SetQoS Service* can not be maintained any longer by a network entity between the receiver and sender. To keep the example clear, the *Negotiation*-, *Start*- and *Stop Phases* are not longer considered. Depending on the capabilities of the available QoS Service Provider protocol the receiver and/or sender's QoS Service Provider is informed about a QoS violation. The *ESL* maps the QoS violation with information about the state of the connection to the upper layer as a *QoSViolation.indication* - passing with the primitive further information about the reason (if this information is available).

It is assumed for the sake of the example, that the sender's side QoS is triggered by a *QoSViolation.indication*. The parameter passed with the primitive informs the QoS Broker about the state of the QoS aware flow, it is assumed that the network entity cannot longer maintain the full-required QoS resulting in changing of the QoS for the specific flow.

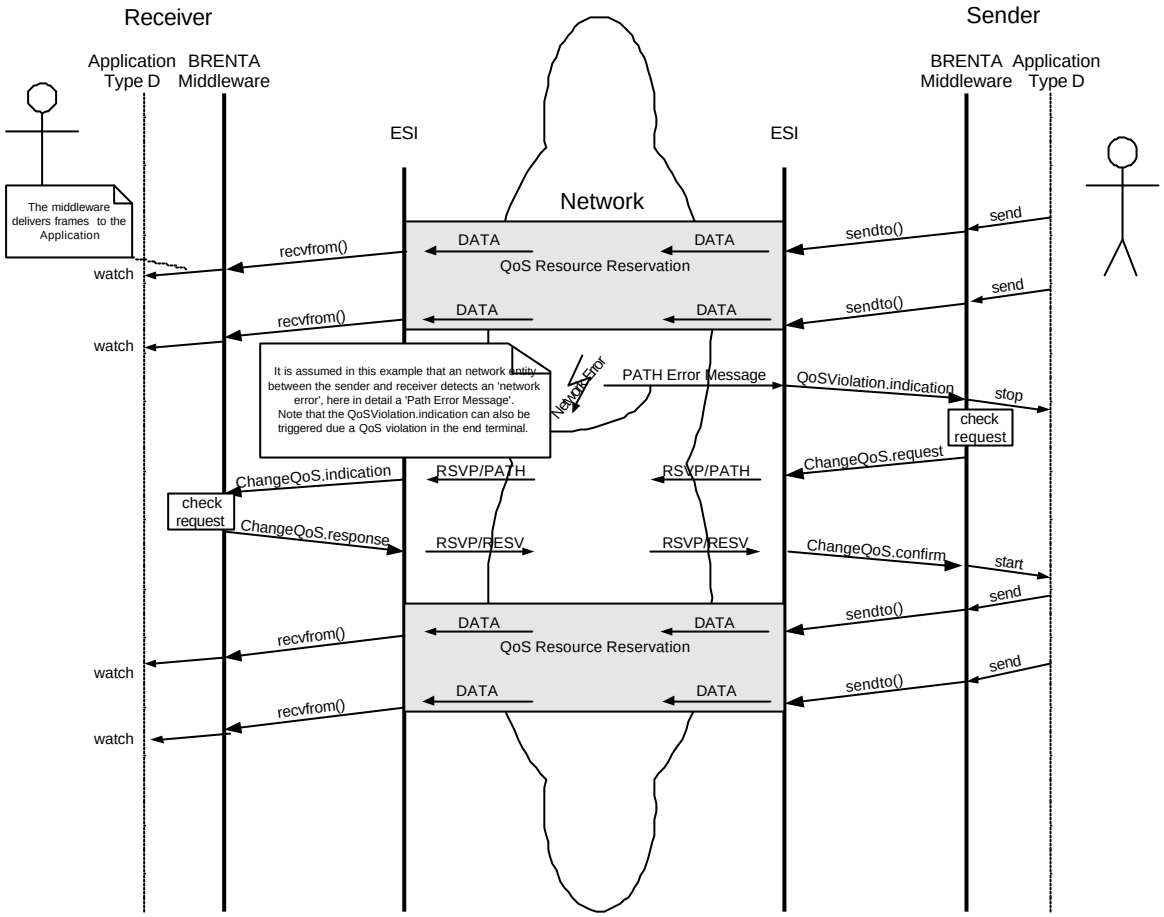


Figure A5-14: Usage of QoSViolation

The *ESI* provides a *Confirmed Service ChangeQoS*. It acts close to the *SetQoS Service* with the difference that the *ChangeQoS Service* can only be applied to a flow associated former with QoS by using *SetQoS*. After successfully changing the QoS for the specific flow the flow is thereafter QoS aware according to the new requested QoS. There might be the case that a sender is still sending packets without considering the fact that the flow might temporarily be not QoS aware - but this depends inherent on the sender's behaviour.

A5.8.9.4 Pragmatically Usage

The previous sections have shown the usage of the *ESI*'s primitive in conjunction with the *BRENTA* middleware. No comments have been spent on how a connection is established. Since this is rather implementation specific it is not considered in detail during the specification of the *ESI*'s services, but to get a clear understanding of how everything plays together it has to be discussed. As shown in Figure A5-16 and
Figure A5-17 a receiver interested in building up a network connection to a sender has to call methods like *socket(AF, Type, Protocol)* resulting in returning a socket descriptor. This descriptor can be used for calling the *connect(sockfd, servaddr, len)* method results either in a) establishing a connection with a corresponding host for connection-oriented protocols or b) to store the correspondent host's address by the process, so that the system knows where to send any future data that the process writes to the socket descriptor. A more detailed description about this process can be found in [A5.9].

Introducing QoS raise different question - should the QoS be handled on the same socket descriptor as the control and data information is handled or should a new QoS specific socket descriptor be introduced as flow parameter for the *ESI*'s services. In the first case the socket descriptor of the *socket()* method has to be used as flow parameter of the *ESI*'s services. A way to specify explicitly that the socket should be QoS aware is the *family parameter* (see [9]) of the socket method, which can be extended by the e.g. *AF_QoS* value - specifying that the socket should be QoS aware - this has the drawback that the non QoS related primitive *socket()* has to be modified.

Considering the latter case the control and data socket can be separated explicitly from a QoS aware socket by introducing a new primitive *QoSSocket* - part of the *ESI.* As parameters can be either passed the socket descriptor returned by the *socket()* function call or primitive specific parameter describing the plain socket plus QoS specific features. Figure A5-15, depicts the primitive call sequence using the *QoSSocket* primitive call. The signature of this primitive can include additional QoS related information for the socket. The *QoSClose* primitive is the complementary to the *close()* method call for QoS unaware sockets. Note for the sake of the example, it is assumed that the *SetQoS Service* is realised in the example API, as a blocking call (see section A5.4, About Implementation).

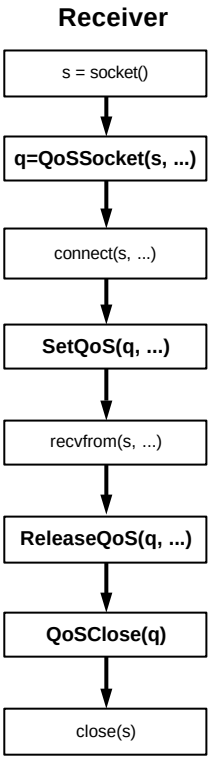


Figure A5-15: QoS aware Socket

A5.9 Brain SI

As mentioned above, the *BRAIN-SI* is a *set* of service interfaces. It is assumed that the interface comprises at least of a *Connection-Oriented-*, *Connection-Less-* and a *Brain-Specific QoS* Service Provider interface. The following sections describe the interface's primitive in more detail. These chapters should be considered more as an examples and proposal how a realisation might look like - in particular for the QoS Service Provider. It is assumed the BRAIN QoS SP looks like a RSVP QoS SP. This is going to change in the future, but it is assumed to have a starting point to show how QoS Mapping can be accomplished.

A5.9.1 BRAIN Connection-oriented Service Provider

This Service Provider comprises mainly out of the BSD ([A5.9]) like *socket*[76] primitives and offers a connection-oriented service to the upper layer. Figure A5-16 summarises the calling sequence and the primitives used to set up such kind of connection.

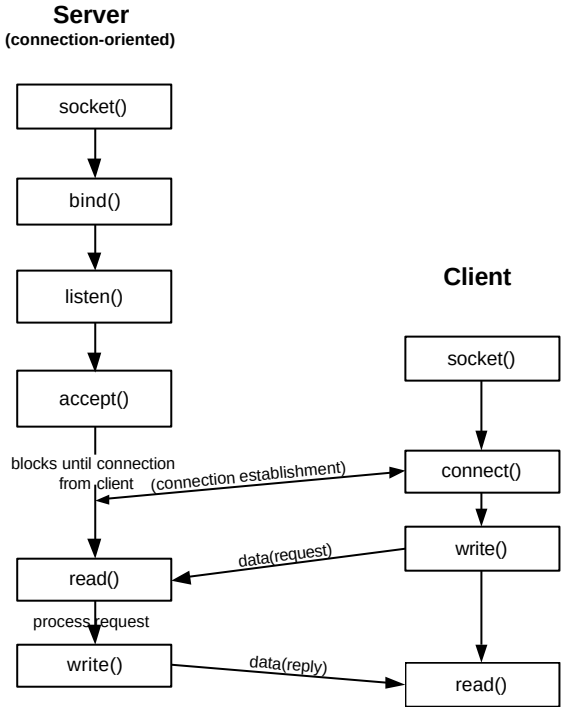


Figure A5-16: Connection-oriented Service Provider

A5.9.2 BRAIN Connection-less Service Provider

As in the connection-oriented way, this *Service Provider* consists out of BSD like primitives using for a user *datagram service*. Primitives and their usage are summarised in [A5.9].

[76] Actually there are two application programming interfaces called socket (Berkley Socket) and TLI (Transport Layer Interface)

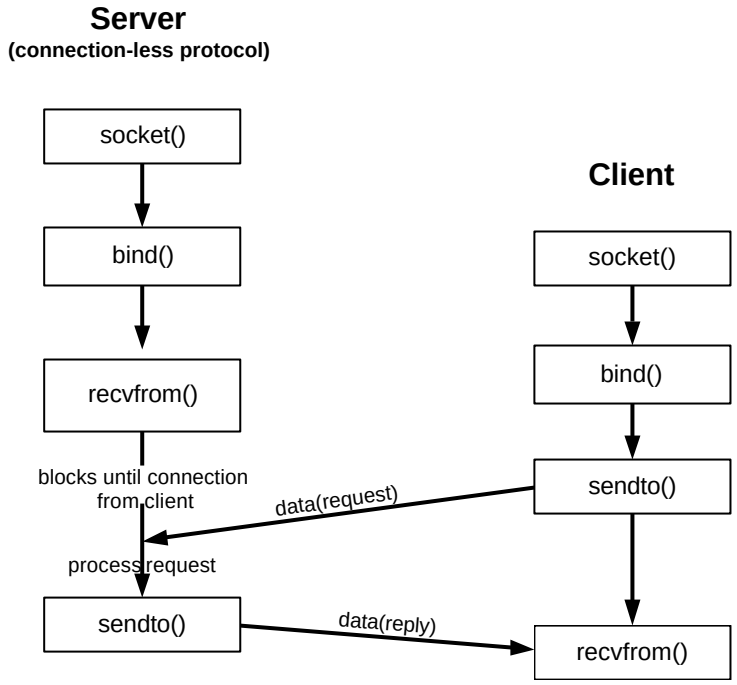


Figure A5-17: Connection-less Service Provider

A5.9.3 BRAIN QoS Service Provider

This section describes a generic interface of a RSVP QoS SP (as explained in [6]) and should be considered as an example. The details of a real interface may be operating system dependent; the following can only suggest the basic functions to be performed.

Session.request primitive		service user -> service provider
associated information	DestAddress , Protocol-Id, DstPort	

Session.confirm primitive		service provider-> service user
associated information	Session-ID	

The *Session.request* primitive initiates RSVP processing for a session, defined by *DestAddress* together with *Protocol-Id* [77] and possibly a port number *DstPort*. If successful, the *Session.confirm* primitive is triggered passing a local session identifier *Session-ID* as parameter, which is used, in subsequent calls.

[77] like TCP or UDP

Sender.request primitive		service user -> service provider
associated information	Session-ID [,Source_Address] [,Source_Port] [,Sender_Template] [,Sender_Tspec] [,Adspec] [,Data_TTL] [,Policy_data]	

A sender uses this primitive to define, or to modify the definition of the attributes of the data flow. The first *Sender.request* primitive call for the session registered as *Session-ID* will cause RSVP to begin sending Path messages for this session; later calls will modify the path information.

The Sender parameters are interpreted as follows:

?? *Source_Address*, is the address of the interface from which the data will be sent. If it is omitted, a default interface will be used. This parameter is needed only on a multihomed sender host.

?? *Source_Port*, is the UDP/TCP port from which the data will be sent.

?? *Sender_Template*, this parameter is included as an escape mechanism to support a more general *definition* of the sender. Normally this parameter may be omitted.

?? *Sender*_Tspec, this parameter describes the traffic flow to be sent, see [A5.7].

?? *Adspec*, this parameter may be specified to initialise the computation of QoS properties along the path; see [A5.7].

?? *Data*_TTL, is the (non-default) IP Time-To-Live parameter that is being supplied on the data packets. It is needed to ensure that Path messages do not have a scope larger than multicast data packets.

?? *Policy*_data, this optional parameter passes policy data for the sender. This data may be supplied by a system service, with the application treating it as opaque.

Sender.indication primitive		service user -> service provider
associated information	Session-ID, Sender_Tspec, Sender_Template [, Adspec] [, Policy_data]	

A *Sender.indication* results from receipt of the first Path message for this session, indicating to a receiver that there is at least one active sender or if the Path State changes. This presents the *Sender_Tspec*, the *Sender_Template*, the *Adspec*, and any *Policy_data* from a Path message (see above for a detailed description).

Reserve.request primitive		service user -> service provider
associated information	session-id, [receiver_address ,] [Policy_data,] style, style-dependent-parms)	

A receiver uses this primitive to make or to modify a resource reservation for the session registered as *Session-ID*. The first *Reserve.request* primitive usage will initiate the periodic transmission of *Resv messages*. A later *Reserve.request* primitive call may be given to modify the parameters of the earlier call (but note that changing existing reservations may result in admission control failures). The optional *receiver_address* parameter may be used by a receiver on a multihomed host (or router); it is the IP address of one of the node's interfaces. The *Policy_data* parameter specifies policy data for the receiver, while the *style* parameter indicates the reservation style(see [A5.6]). The rest of the parameters depend upon the style; generally these will be appropriate flowspecs and filter specs.

Reserve.indication primitive	service user -> service provider

associated information	Session-Id, Style, Flowspec, Filter_Spec_list [, Policy_data]

A *Resverve.indication* primitive is triggered by the receipt of the first RESV message, or by modification of a previous reservation state, for this session. Here *Flowspec* will be the effective QoS that has been received.

Release.request primitive	service user -> service provider
associated information	Session-Id

This primitive removes RSVP state for the session specified by Session-Id. The node then sends appropriate tear down messages and ceases sending refreshes for this Session-Id.

PathViolation.indication primitive	service provider -> service user
associated information	Error_code , Error_value , Error_Node , Sender_Template [, Policy_data_list]

A *PathViolation.indication* primitive indicates an error in sender information that was specified in a *Sender.request* primitive call. The *Error_code* parameter will define the error, and *Error_value* may supply some additional (perhaps system-specific) data about the error. The *Error_Node* parameter will specify the IP address of the node that detected the error. The *Policy_data_list* parameter, if present, will contain any *Policy_data* objects from the failed Path message. Detailed information about error codes and their values can be found in RFC2205 [A5.6], APPENDIX B. Error Codes and Values.

ResvViolation.indication primitive	service provider -> service user
associated information	Error_code , Error_value , Error_Node , Error_flags, Flowspec, Filter_spec_list [, Policy_data_list]

A *ResvViolation.indication* primitive indicates an error in a reservation message that was specified in a *Reserve.request* primitive call. The *Error_code* parameter will define the error and Error_value may supply some additional (perhaps system-specific) data. The *Error_Node* parameter will specify the IP address of the node that detected the event being reported. There are two Error_flags a) *InPlace*, which may be on for an admission control failure, to indicate that there, was, and is, a reservation in place at the failure node. This flag is set at the failure point and forwarded in *ResvErr* messages b) *NotGuilty* which may be on for an Admission Control failure, to indicate that the flowspec requested by this receiver was strictly less than the flowspec that got the error. This flag is set on the receiver side. *Filter_spec_list* and *Flowspec* will contain the corresponding objects from the error flow descriptor. *List_count* will specify the number of *FILTER_SPECS* in *Filter_spec_list*. The Policy_data_list parameter will contain any *POLICY_DATA* objects from the *ResvErr* message. Detailed information about error codes and their values can be found in RFC2205 [A5.6], APPENDIX B. Error Codes and Values.

Summary of the important Parameters used for the Sender and Receiver service

Section A5.11 shows how the *ESI* primitives and their QoS Parameters can be mapped to a RSVP like QoS Service Provider's primitives and parameters introduced above. The essential parameters used by a RSVP QoS Service Provider are summarised below to get a clear understanding how the mapping of *QoS Parameters* can be accomplished - for a more detailed description see [A5.7].

RSVP Parameter	Description

Sender_TSPEC	Carries the traffic specification (Sender Traffic Specification) generated by a sender. It is transported unchanged through the network, and delivered to both intermediate nodes and receiving applications.
ADSPEC	Carries information which is generated at either data sources or intermediate network elements, is flowing downstream towards receivers, and may be used and updated inside the network before being delivered to receiving applications.
Receiver_TSpec & RSpec comprised in Flowspec	RSVP Receiver Flowspec object carries reservation request (Receiver_TSpec and RSpec - Reservation Specification) information generated by data receivers. The information in the Flowspec flows upstream towards data sources.

Table A5-2: RSVP QoS related Parameters

Table A5-2 summarises the important RSVP Parameter - especially the *Sender_TSPEC*, *Receiver_TSpec* and *RSpec*. How they are used between a sender and receiver is depicted in Figure A5-18.

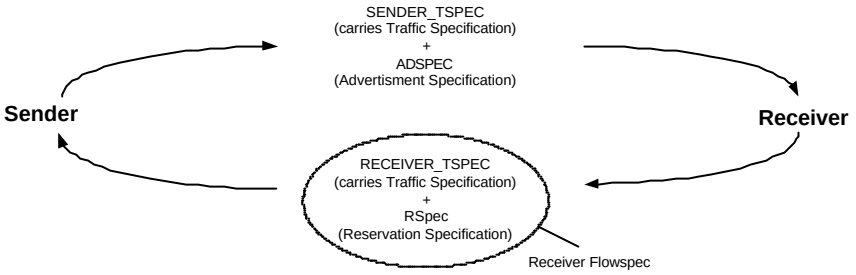


Figure A5-18: RSVP QoS related Parameters and how they play together

A5.10 QoS Parameters for the ESI

In this section we propose QoS parameters that are used in the primitives defined above. Note that these parameters have to be generic in order to support all kind of networks. Of course, some network types do not support certain QoS enforcement mechanisms and thus cannot provide certain guarantees. A proper reaction would be to throw an indicating that the current Service Provider cannot support the specific parameter constraints. However, it might be very likely possible that another Service Provider, i.e. a Service Provider that uses network technology that can provide tight QoS guarantees, can invoke enforcement mechanisms to guarantee the specific parameter set.

By specifying generic QoS parameters the application code needs not to be changed if the application is running on different platforms with different QoS Service Providers. It is then the task of the *ESL* to map generic QoS parameters to the used QoS SP's QoS parameters.

By specifying QoS parameters, QOS-aware applications may invoke, modify, or remove quality of service settings for a given flow. We adopt the well-known concept of a *Flowspec* to invoke proper QoS treatment on a given flow. In order to provide more freedom to the QoS SP and to reduce the amount of QoS violation messages, we introduce the concept of *parameter intervals*, where appropriate.

This concept is motivated by the fact that not all applications require one fixed amount of resources during the lifetime. For example, an adaptive video application may be satisfied with an end-to-end delay between 150 and 250 ms and a sustainable bandwidth between 1 Mbps and 1.25 Mbps. However such an application would never send more than 1.2 Mbps to its peer.

Therefore, we specify QoS parameters as intervals, where appropriate. For each such parameter the lower bound is denoted as *minimum_acceptable_limit* and the upper bound as *desirable_ limit*. The QoS SP should then invoke proper mechanisms to guarantee (how tight this guarantee is depends on the service type - guaranteed, best effort or controlled load) the minimum acceptable limit for each parameter during the lifetime of the flow. Whenever the QoS SP detects that more resources are available, it can book resources up to the desirable limit to the given flow. By specifying such intervals, the QoS SP may start at any appropriate working point that is inside the interval specification given by the QoS parameters. The QoS SP then tries to stay with the working point inside the bounding box as long as possible.

Note that for some parameters it does not make sense to specify intervals (like service type). The application might also request tighter intervals by setting *minimum_acceptable_limit = desirable_ limit* for the respective parameter.

A5.10.1 QoS Parameters

The QoS parameter provides the means by which QOS-enabled applications can specify quality of service parameters for sent or received traffic on a particular flow. We explicitly distinguish between sending and receiving traffics QoS characteristics because most applications use show asymmetric traffic behaviour. The QoS parameter comprises the following parameters:

SendingFlowspec

Specifies QoS parameters for the sending direction of a particular flow. *SendingFlowspec* is sent in the form of a *FLOWSPEC* parameter specified in the next section.

ReceivingFlowspec

Specifies QoS parameters for the receiving direction of a particular flow. *ReceivingFlowspec* is sent in the form of a FLOWSPEC parameter specified in the next section.

ServiceProviderSpecific [OPTIONAL]

This parameter can provide additional provider-specific QoS parameters to the QoS SP for a given flow. Alternatively, the flowspec could provide an extensibility mechanism, i.e. a set of rules for adding additional information in a structured manner (e.g. by providing for a given additional information item, an item ID, indicators specifying what treatment should be applied to the item if the corresponding SP functionality is missing or not compatible - e.g. ignore - the length of the description, and the description).

Note that most applications can fulfil their QoS requirements without using the *ServiceProviderSpecific* extensions. However, if the application must provide information not available with the QoS parameters defined in this chapter, the *ServiceProviderSpecific* extensions allows the application to provide additional parameters for the QoS Service Provider.

Note also not all parameters have to be specified for a QoS request to the *ESI*. If the *ESL* cannot map the requested QoS to the used QoS Service Provider a specific error message is return. If for a sending flow also a receiving *flowspec* is defined it is simply be ignored since a flow is a uni-directional data stream. The same is valid for the opposite case. But if for a sending flow no specific *SendingFlowspec* is defined an error message is returned.

A5.10.2 FlowSpec

The *FlowSpec* provides the basis for QoS contracts. In addition to best effort traffic, where applications only tell the network where to deliver the packets, QoS enabled applications have to tell the network more precisely its requirements for QoS, the type of service the application requires and the amount of traffic that the app is going to inject into the network. The application might provide qualitative information like "use a controlled-delay service" or quantitative information like "I need a maximum delay of 150 ms". In addition to the description, what the application requires additional information what the application is going to inject into the network helps the service provider in planning its resources.

Within the *FlowSpec*, the application is specifying its requirements for the given flow, i.e. the *FlowSpec* is the set of information we provide to the network. It is then up to the *ESL* and the used QoS SP to interpret these parameters and to take proper actions (e.g. traffic control mechanisms and/or signalling) to enforce the QoS contract.

A *FlowSpec* consists of:

Token Bucket Model Parameters (see [A5.5])

TokenRate	Specifies the permitted rate at which data can be transmitted over the life of the flow.
TokenBucketSize	The maximum amount of credits a given direction of a flow can accrue, regardless of time.
PeakBandwidth	The upper limit on time-based transmission permission for a given flow sometimes considered a burst limit. PeakBandwidth restricts flows that may have accrued a significant amount of transmission credits, or tokens from overburdening network resources with one-time or cyclical data bursts, by enforcing a per-second data transmission ceiling. Some intermediate systems can take advantage of this information, resulting in more efficient resource allocation. Note that the PeakBandwidth must be greater than or equal to TokenRate

Latency	Maximum acceptable delay between transmission of a bit by the sender and its receipt by one or more intended receivers. The precise interpretation of this number depends on the level of guarantee specified in the QoS request. Latency is expressed as an interval (minimum_acceptable_limit and desirable_ limit in microseconds). Specifying an interval for their Latency may satisfy elastic or adaptable applications. The QoS SP invokes appropriate mechanisms that depend on the service type to enforce the minimum_acceptable_limit over the lifetime of the flow. If this cannot be maintaining, a QoS violation is signalled to the service user. If enough resources are available, the QoS SP may provide resources that allow enforcing desirable_ limit. Note, that the lower the value in latency, the higher the resource requirements are.

DelayVariation/Jitter Difference between the maximum and minimum possible delay a packet will experience. Applications use *DelayVariation* to determine the amount of buffer space needed at the receiving end of the flow, in order to restore the original data transmission pattern. *DelayVariation* is expressed as an *interval* (*minimum_acceptable_limit* and *desirable_ limit* in microseconds).

Specifying an interval for the required DelayVariation may satisfy elastic or adaptable applications. The QoS SP invokes appropriate mechanisms that depend on the service type to enforce the *minimum_acceptable_limit* over the lifetime of the flow. If this cannot be maintaining, a QoS violation is signalled to the service user. If enough resources are available, the QoS SP may provide resources that allow enforcing *desirable_ limit*. Note, that the lower the value in latency, the higher the resource requirements are.

Service Types

BESTEFFORT	Specifies that the Resource Provider should use the *FLOWSPEC* as a service quality guideline, and make reasonable efforts to maintain the level of service requested, without making any guarantees on packet delivery.
CONTROLLED LOAD	Provides an end-to-end quality of service that closely approximates transmission quality provided by best-effort service, as expected under *unloaded conditions* from the associated network components along the data path. Applications that use *CONTROLLEDLOAD* may therefore assume that the network will deliver a very high percentage of transmitted packets to their intended receivers; in other words, packet loss will closely approximate the basic packet error rate of the transmission medium. Transmission delay for a very high percentage of the delivered packets will not greatly exceed the minimum transit delay experienced by any successfully delivered packet.
GUARANTEED	Initiates QoS enforcement mechanisms within the service provider that isolates a given flow from the effects of other flows (as possible). This isolation guarantees the ability to transmit data at *TokenRate* (*minimum_acceptable_limit*) for the duration of the connection. However, if the corresponding end-node transmits data faster than *TokenRate*, the network may delay or discard the excess traffic based on a policing/dropping behaviour. If *TokenRate* (*minimum_acceptable_limit*) is not exceeded over time, *Latency* (*minimum_acceptable_limit*) is also guaranteed. If enough resources are available, the service provider may support the higher interval bounds. *GUARANTEED* is designed for applications that may require a deterministic quality of service but would not benefit from better service (such as real-time control systems).

MaxSduSize Specifies the maximum packet size permitted or used in the traffic flow. *MaxSduSize* is expressed in bytes.

MinimumPolicedSize Specifies the minimum packet size for which the requested QoS will be provided. *MinimumPolicedSize* is expressed in bytes.

A5.10.3 Service Provider specific Information [optional]

As mentioned in the preliminary sections the QoS parameter structure can contain QoS Service Provider specific information. This additional information is passed to the *ESI* which itself manage the data and passed it if appropriate to the currently used QoS Service Provider. The QoS Service Provider specific information can be seen as a hook concept, allowing an application to pass QoS Service Provider specific information. It has to be noted, that there is absolute no guarantee that the parameters are applied in any way.

A Service Provider specific information may contain:

PacketLossRate Specifies the packet loss rate that is desirable for the given service. *PacketLossRate* is expressed as an *interval* of floating numbers (*minimum_acceptable_limit* and *desirable_ limit* in percentage of packets lost).

Specifying an interval for their PacketLossRate may satisfy elastic or adaptable applications. If the used QoS SP provides this parameter, it can invoke appropriate mechanisms that depend on the service type to enforce the *minimum_acceptable_limit* over the lifetime of the flow. If enough resources are available, the QoS SP may provide resources that allow enforcing *desirable_ limit.*

ShapeDiscardMode

Specifies the requested behaviour of a Packet Shaper used by traffic control mechanisms if traffic control is implemented. Values are:

NONCONF_BORROW	Instructs a Packet Shaper to *borrow* remaining available resources *after* all higher priority flows have been serviced. If the *TokenRate* is specified for this flow, packets that exceed the value of *TokenRate* will have their priority denoted to less than *SERVICETYPE_BESTEFFORT*, as defined by *service type*.
NONCONF_SHAPE	Instructs a Packet Shaper to retain packets until network resources are available to the flow in sufficient quantity to make such packets conforming. (For example, a 100K packet will be retained in the Packet Shaper until 100K worth of credit is accrued for the flow, allowing the packet to be transmitted as conforming). *TokenRate* must be specified if using *TC_NONCONF_SHAPE*.
NONCONF_DISCARD	Instructs the Packet Shaper to discard all non-conforming packets.

PacketDropPriority Specifies the requested dropping priority for the given flow. This parameter may be used to prioritize between different flows that show the same behaviour. *PacketDropPriority* implicitly determines the drop behaviour that should be applied to the given flow. The lower the *PacketDropPriority*, the lower the local dropping possibility. *PacketDropPriority* is specified as an *interval* of byte numbers (*minimum_acceptable_limit* and *desirable_ limit*). The lowest *PacketDropPriority* is 0, whereas the highest is 255.

As an example, an audio flow might require a lower dropping possibility than a video flow.

Specifying an interval for their *PacketDropPriority* may satisfy elastic or adaptable applications. The *QoS SP* invokes appropriate mechanisms that depend on the service type to enforce the *minimum_acceptable_limit* over the lifetime of the flow. If enough resources are available, the QoS SP may provide resources that allow enforcing *desirable_ limit.*

PacketForwardPriority

Specifies the requested forwarding priority for the given flow. *PacketForwardPriority* implicitly determines the queuing behaviour that should be applied to the given flow. The higher the *PacketForwardPriority*, the lower the local queuing delay. *PacketForwardPriority* is specified as an *interval* of byte numbers (*minimum_acceptable_limit* and *desirable_ limit*). The lowest *PacketForwardPriority* is 0, whereas the highest is 255.

As an example, an audio flow might require a higher packet forwarding priority than a video flow.

Specifying an interval for their *PacketDropPriority* may satisfy elastic or adaptable applications. The QoS SP invokes appropriate mechanisms that depend on the service type to enforce the *minimum_acceptable_limit* over the lifetime of the flow. If enough resources are available, the QoS SP may provide resources that allow enforcing *desirable_ limit. PacketDropPriority implicitly* determines the queuing mechanisms applied for hand over situations for the given flow. Together with *PacketLossRate* it determines what error control mechanisms should be applied, if available.

Cost

Including cost into a QoS specification would enable application programmers to include cost issues into the QoS contract. As a it was decided that this parameter shall be regarded as *spare container* parameter, to be specified further in a later project.

A5.11 QoS and Primitive Mapper

One basic idea of the *ESI* is to give support in writing QoS aware applications, independent from the available QoS Service Providers. Therefore functionality is defined mapping the *ESI's* primitives and their parameters onto the appropriate and usable QoS Service Provider's primitives and parameters. The Mappers are dependent on *ESL's* registered QoS Service Provider. If the usability of the QoS SP change e.g. due handover, either the Mapper have to be exchanged or they have to be informed that they have to map henceforth between the *ESI* and the new usable QoS SPs. The adaptation of the Mapper must be done by the *ESL*, which therefore has to be informed about the change of usable QoS SP from underlying layers.

As already described in section A5.1, the *ESL* provides at least two entities the QoS and Primitive Mapper respectively. The following two sections demonstrate how such Mapper can look like. It is started with a Primitive Mapper followed by a QoS Mapper. These sections show only the logical mapping of the primitives and QoS Parameters. How such Mappers are realised is rather implementation specific.

Also details considering setting up a session, if provided by the available QoS Service Provider, is left open for further detailed implementation specification. Note that the Mappers are explained each for their own, but at the end they have to play together - for example, during mapping the *SetQoS* service primitive the specified QoS parameters have to be considered and mapped in an appropriate way. The QoS Parameter can also influence the primitive Mapper at all. If the requested service type is like *Best Effort* it might be that the *SetQoS* service primitives are mapped to different kind of primitives as in the case of requesting a service type like *Guaranteed Load*.

The scope of this section is to show how a mapping between explicit end-to-end signalling services, supported by the ESI, and an appropriate QoS Service Provider as introduced in chapter A5.9.3 can be done. Mapping of non explicit end-to-end signalling services like *AssignQoS* is not considered in this section.

A5.11.1 Primitive Mapper

The Primitive Mapper is in charge of mapping the *ESI primitives* to usable/available QoS Service Provider and vice versa. Based on the design decisions made in chapter A5.7 it has to be distinguished between services related to explicit end-to-end signalling like the *SetQoS* and *ChangeQoS* services and services which are explicitly not using any end-to-end signalling like the *AssignQoS* service.

In the following it is described how the *SetQoS* and *ChangeQoS* and their associated *SetQoS-ChangeQoSViolation* service primitives can be mapped. Therefore it is assumed that an explicit end-to-end signalling QoS Service Provider is available, for the sake of the example it is assumed that a RSVP QoS SP is available and usable by the ESL. The RSVP primitives introduced in chapter A5.9.3 are used for describing the mapping.

A5.11.1.1 Mapping of SetQoS and SetQoSViolation service primitives

The exact mapping of the *SetQoS* and *SetQoSViolation Service* primitives is straightforward and is depicted in Table A5-3 and Table A5-4. Note that *PathViolation.indication* and *ResvViolation.indication* is mapped to *SetQoSViolation.indication* only during establishing of a QoS aware flow. Further *Path-* or *ResvViolation.indication* are mapped to the *QoSViolation.indication* primitive. Figure A5-19 summarises the mapping in a time sequence diagram including the messages sent between the participating sides.

ESI	QoS SP (RSVP)	Direction
SetQoS.request	Sender.request	ESI->(RSVP) QoS SP
SetQoS.indication	Sender.indication	ESI<-(RSVP) QoS SP
SetQoS.response	Reserve.request	ESI->(RSVP) QoS SP
SetQoS.confirm	Reserve.indication	ESI<-(RSVP) QoS SP

Table A5-3: Mapping of SetQoS Service Primitives

ESI	QoS SP (RSVP)	Direction
SetQoSViolation.indication	PathViolation.indication	(RSVP) QoS SP->ESI
SetQoSViolation.indication	ResvViolation.indication	(RSVP) QoS SP->ESI

Table A5-4: Mapping of SetQoSViolation Service Primitives

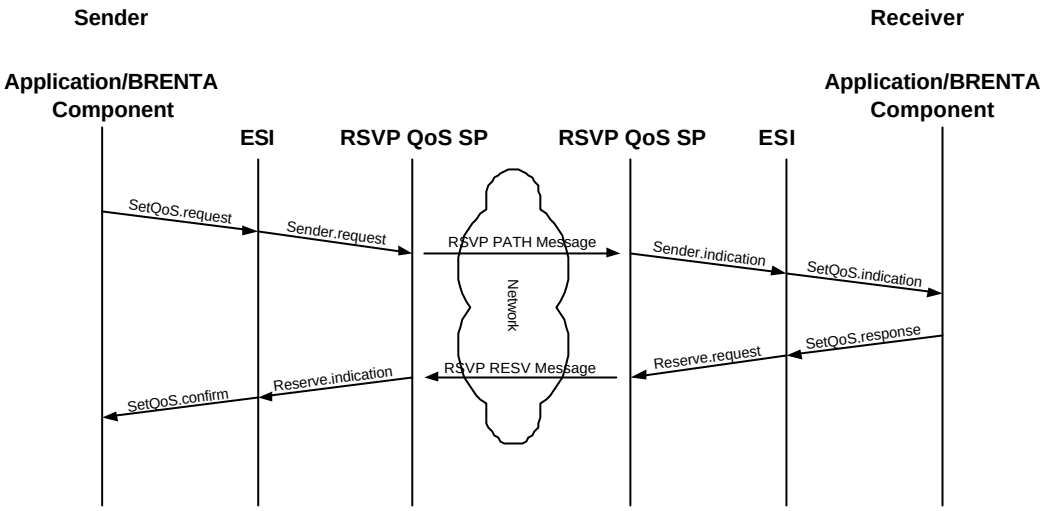


Figure A5-19: Mapping of SetQoS Service

A5.11.1.2 Mapping of ChangeQoS and ChangeQoSViolation service primitives

The *ChangeQoS* primitives can be mapped in the same way as the *SetQoS* primitives due the fact these RSVP QoS SP primitives are also used in the same way for changing the QoS properties of a QoS aware flow. Details of the mapping are depicted in Table A5-5. Note that *PathViolation.indication* and *ResvViolation.indication* is mapped to *ChangeQoSViolation.indication* only during the change of the properties of a QoS aware flow. *Path* or *ResvViolation.indication* received before/after a change of the QoS aware flow, are mapped to the *QoSViolation.indication* primitive. Figure A5-20 summarises the mapping in a time sequence diagram including the message sent between the participating sides.

ESI	QoS SP (RSVP)	Direction
ChangeQoS.request	Sender.request	ESI-> QoS SP (RSVP)
ChangeQoS.indication	Sender.indication	ESI<- QoS SP (RSVP)
ChangeQoS.response	Reserve.request	ESI-> QoS SP (RSVP)
ChangeQoS.confirm	Reserve.indication	ESI<- QoS SP (RSVP)

Table A5-5: Mapping of ChangeQoS Service Primitives

ESI	QoS SP (RSVP)	Direction
ChangeQoSViolation.indication	PathViolation.indication	(RSVP) QoS SP->ESI
ChangeQoSViolation.indication	ResvViolation.indication	(RSVP) QoS SP->ESI

Table A5-6: Mapping of ChangeQoSViolation Service Primitives

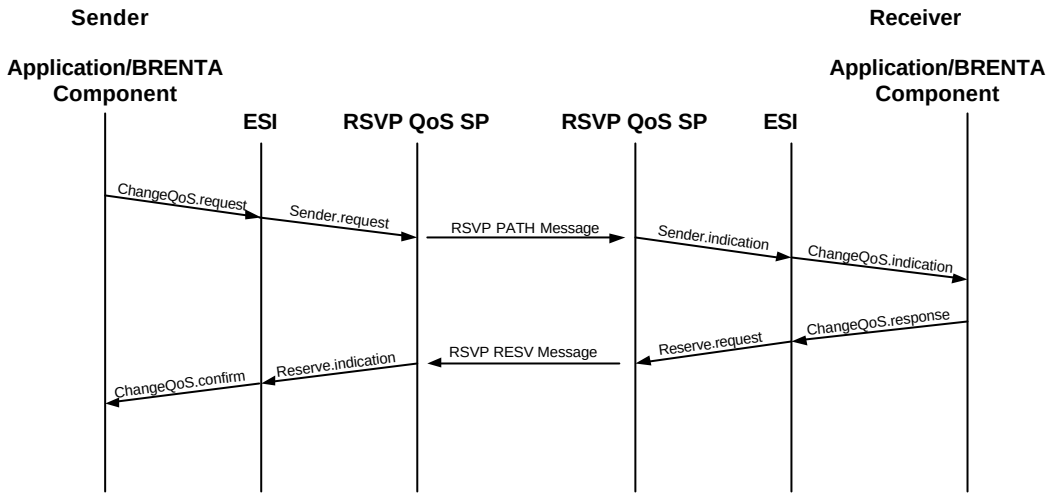


Figure A5-20: ChangeQoS Mapping

A5.11.1.3 Error Cases

During sending *Path* and *Resv Messages* between the participating hosts any network entity can generate an error event due sending *Path* or a *Resv Error Messages*. Thereby a *Path Error Message* is generated in reply to a send *Path Message* and a *Resv Error Message* is sent in reply to a *Resv Message*.

From a QoS SP point of view *PathViolation.indication* primitive indicates an error in sender information that was specified in a *Sender.request* primitive call. A *ResvViolation.indication* primitive indicates an error in a reservation message that was specified in a *Reserve.request* primitive call.

ESI	QoS SP (RSVP)	Direction
QoSViolation.indication	PathViolation.indication	(RSVP) QoS SP->ESI
QoSViolation.indication	ResvViolation.indication	(RSVP) QoS SP->ESI

Associated with the *Path* and *ResvViolation.indication* primitive information about the error is passed as an argument. The information about the error, also termed *Error code* (see Appendix [A5.7]), can be mapped 1:1 to the *type* parameter of the *QoSViolation.indication* primitive. In addition to this information the *ESL* has to add information about the state of the QoS aware flow. This information can be deduced from the *Error code* if no further information[78] is available. For example, if the flow is already QoS aware and the *ResvViolation.indication* primitive is called with an *Error Code = 01 Admission control failure - Reservation request was rejected by admission control due to unavailable resource* - it can be deduced that the flow is not longer completely[79] QoS aware.

A5.11.2 QoS Mapper

The QoS Mapper is in charge of mapping the *ESI* QoS Parameters (see chapter A5.10) to the QoS Service Provider supporting QoS related parameters. Due the usage of a RSVP QoS Service Provider the mapping consists mainly between the *ESI*'s QoS Parameter and the RSVP QoS Service Provider's *TSpec* (T representing traffic, so *TSpec* means *Traffic Specification*, either for sender or receiver) and *RSpec* (R representing reservation, so *RSpec* means Reservation Specification). It has to be distinguished between the sender and receiver's side. On the sender's side the QoS Parameter can be directly mapped to the Sender's *TSpec* parameter, used in the *Sender*'s service primitives whereas on receiver side the QoS Parameters has to be mapped to the Receiver's *TSpec* and *RSpec* depending on the client's requested service type[80]. For more details see [A5.7], a short introduction can also be found in chapter A5.9.3 .

[78] Context information deduced from the current processing of the request/response.

[79] Note that the failure is generated by **one** specific network entity, it might be that all the other network entities between the sender and receiver could support a QoS aware flow.

[80] Best Effort, Control Load or Guaranteed Load

A5.11.2.1 Mapping ESI QoS Parameter to RSVP Sender TSpec and vice versa

Table A5-7 shows a mapping of the ESI's QoS Parameter to RSVP *Sender TSpec* parameter. The mapping is applied during mapping the RSVP *Sender's* primitives. This means that the *ESI* QoS Parameters passed with a *SetQoS.request* primitive call (mapped to *Sender.request*) are directly mapped to the RSVP *Sender.request TSpec* parameters according Table 5. The same is valid during mapping the RSVP *Sender.indication* primitive to the ESI *SetQoS.indication* primitive whereby the *TSpec* is mapped 'back' to the *SetQoS.indication* 's QoS Parameters. (The same procedure is valid for *ChangeQoS* service primitives QoS Parameter mapping).Note, that the *Latency* and *DelayVariation* is simply ignored or left empty in this case - in contrast to the mapping on the receiver side (see below). Figure A5-21, at the end of this section, depicts how the parameter mapping relates to the primitive mapping.

QoS Parameter	RSVP Sender TSpec	Direction
TokenRate	TokenBucketRate	(RSVP) QoS SP <-> ESI
TokenBucketSize	TokenBucketSize	(RSVP) QoS SP <-> ESI
PeakBandwidth	PeakRate	(RSVP) QoS SP <-> ESI
MinimumPolicedSize	MinimumPolicedUnit	(RSVP) QoS SP <-> ESI
MaxSduSize	MaximumPacketSize	(RSVP) QoS SP <-> ESI
Latency		(RSVP) QoS SP <-> ESI
DelayVariation		(RSVP) QoS SP <-> ESI

Table A5-7: Mapping of ESI QoS Parameters to RSVP Sender TSpec

A5.11.2.2 Mapping ESI QoS to RSVP Receiver's Flowspec (TSpec and RSpec)

Table A5-8 and Table A5-9 outlines a mapping between the *ESI*'s QoS Parameter and RSVP *Receiver Flowspec* parameter comprising Receiver's *TSpec* and *RSpec*. The mapping is applied during mapping the RSVP *Receiver* primitives. This means that *ESI* QoS Parameters passed with a *SetQoS.response* primitive call (mapped to *Receiver.request*) are mapped to the RSVP *Receiver.request TSpec* and *RSpec* parameters respectively. The same is valid during mapping the RSVP *Receiver.indication* primitive to the *ESI SetQoS.confirmation* primitive whereby the *TSpec* and *RSpec* is mapped 'back' to the *SetQoS.confirmation* 's QoS Parameters. (The same procedure is valid for *ChangeQoS* service primitives QoS Parameter mapping). Figure A5-21, at the end of this section depicts how the parameter mapping relates to the primitive mapping.

QoS Parameter	RSVP Sender Tspec	Direction
TokenRate	TokenBucketRate	(RSVP) QoS SP <-> ESI
TokenBucketSize	TokenBucketSize	(RSVP) QoS SP <-> ESI
PeakBandwidth	PeakRate	(RSVP) QoS SP <-> ESI
MinimumPolicedSize	MinimumPolicedUnit	(RSVP) QoS SP <-> ESI
MaxSduSize	MaximumPacketSize	(RSVP) QoS SP <-> ESI
Latency		(RSVP) QoS SP <-> ESI
DelayVariation		(RSVP) QoS SP <-> ESI

Table A5-8: Mapping of ESI QoS Parameters to RSVP Receiver TSpec and RSpec

The RSVP *RSpec* specifies requested QoS parameters and is used by the receiver in *RESV messages* to transmit requested reservation parameters only when service of type *Quaranteed* is specified by the application. *RSpec* consists of a *Rate* and *SlackTerm* parameter (see [RFC2210]). Mapping to the *RSpec* is depicted in Table A5-9.

QoS Parameter	RSVP Receiver RSpec	Direction
TokenRate and DelayVariation	Rate is copied from TokenRate *SlackTerm* is copied from *DelayVariation*. The L*atency* parameter is ignored.	(RSVP) QoS SP <-> ESI
DelayVariation and Latency	*Rate* parameter of *RSpec* calculated based on DelayVariation and Latency	(RSVP) QoS SP <-> ESI

Table A5-9: Mapping of ESI QoS Parameters to RSVP Receiver TSpec and RSpec

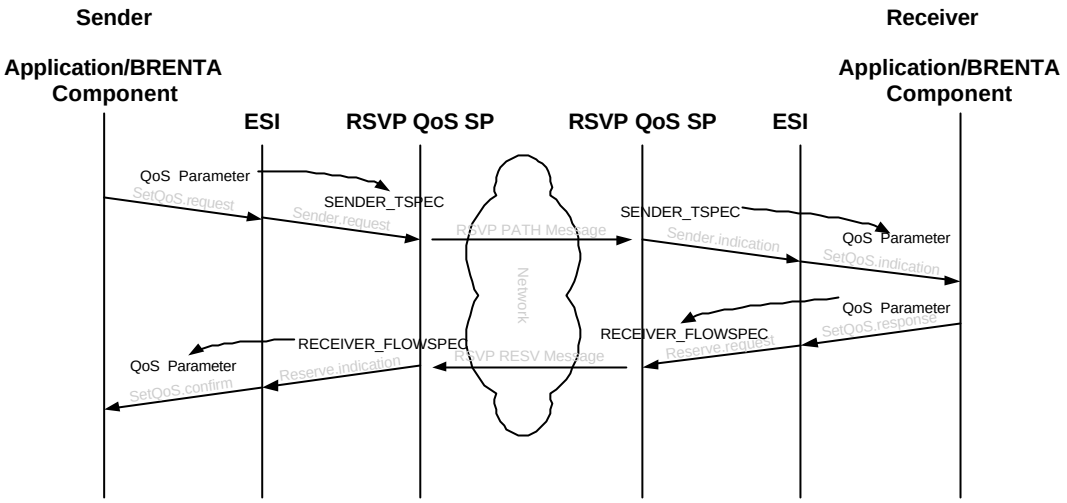


Figure A5-21: Mapping of QoS Parameter to Sender TSpec and Receiver FlowSpec

A5.12 Support of QoS unaware Application

As depicted in Figure A5-1 the *ESI* supports both, QoS aware Applications and QoS unaware Applications also called legacy Application. The focus of this chapter is on legacy Application or using BRENTA [BRENTA] terminology - on *Type-A* applications. How the *ESI* supports legacy Application is described in this chapter.

A5.12.1 What is a legacy Application?

Before going into detail how a legacy Application can be supported, it has to be worked out what a legacy Application is. Requirements have to be elaborated what a QoS aware Application is and from these the support of a QoS unaware Application can be deduced. To find out the requirements a receiver/sender scenario is investigated in the following.

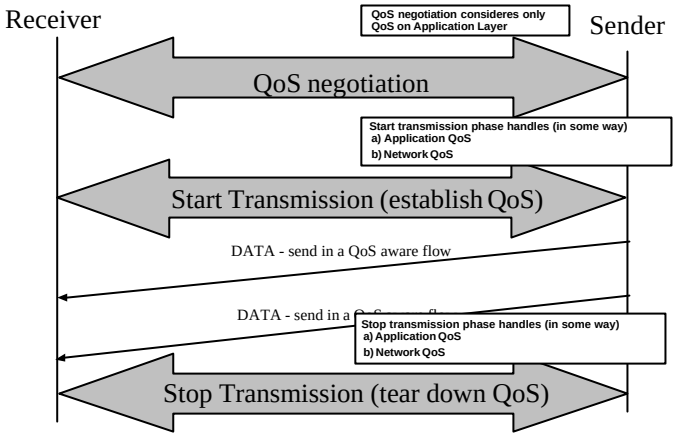


Figure A5-22: What is a QoS aware Application

As depicted in Figure A5-22, a QoS aware Application, either in the role of a Sender or Receiver, a) can participated in a QoS negotiation phase b) has to participate in a *Start Transmission* phase and c) has to participate in the *Stop Transmission* phase. During the *Start* and *Stop Transmission* phase the necessary network QoS for a QoS aware flow has to be set-up. A QoS unaware Application has to be supported in a way that the corresponding side assumes that a QoS capable corresponding peer is available. If the Sender/Receiver scenario is analysed from a QoS Aware and Legacy Application in more detail four different cases can be considered as depicted in Table A5-10.

Case	Receiver	Sender
1)	QoS Aware Application	QoS Aware Application
2)	Legacy Application	QoS Aware Application
3)	QoS Aware Application	Legacy Application
4)	Legacy Application	Legacy Application

Table A5-10: Analyse the Sender/Receiver Scenario

Case one has been already elaborated in conjunction with Work Group [A5.1]. The focus here is on the other cases. It has to be noted that the support of QoS unaware Application makes only sense in specific scenarios where the supported legacy Application is *friendly* in a way that it can be supported with QoS by a third party application. It is out of discussion, that there are scenarios where the support of legacy Application can not be accomplished in a meaningful way - hence there is no aim to support any kind of Legacy Applications. The focus is preferentially on considering specific scenarios and based on the results a meaningful support of Legacy Application should be achieved.

Due the fact that a Legacy Application is not QoS aware a third party component on the mobile terminal has to take over and slips into the role of a QoS proxy for legacy Applications. An Application doing that is termed *Configurator* (see also [A5.3]). It might have a Graphical User Interface allowing user-friendly support of legacy Application. Due the fact that the details of how a *Configurator* looks like are rather implementation specific they are not discussed in the scope of this document.
Note also, that the forth case (see Table A5-10) can be derived from the second - and third case, so for the sake of analysing the scenario the fourth case is skipped.

A5.12.2 Consider the support of Receiver QoS unaware Applications

If the case that a receiver is a QoS unaware Application is considered the *Configurator* has to take over the task of handling QoS for the Legacy Application. As mentioned above the *Configurator* has at least to participate in the *Start Transmission* and *Stop Transmission* phase to establish a QoS aware flow.

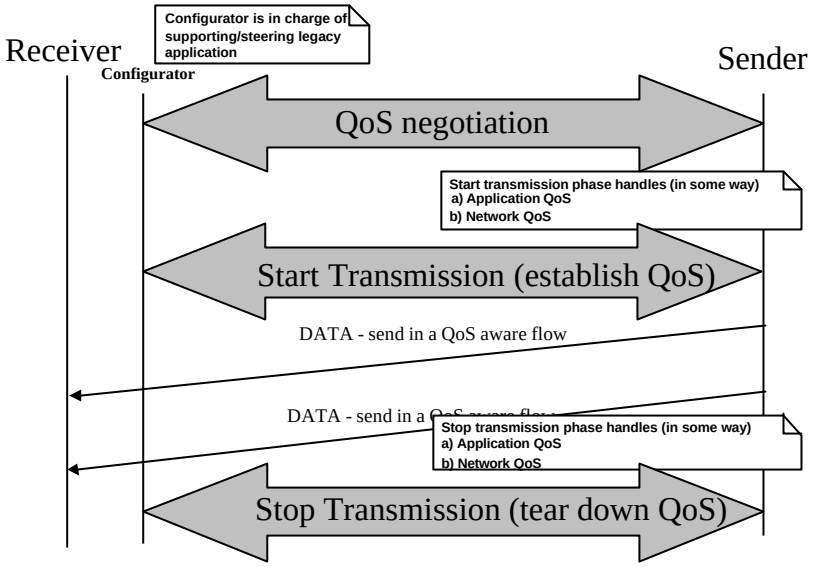


Figure A5-23: Receiver QoS unaware Application

The exact behaviour of the *Configurator* depends inherent on the used QoS Service Provider - mainly between Sender-initiated (like YESSIR) and Receiver-initiated (like RSVP) ones. Is the Sender-initiated case is considered, the *Configurator* is in charge of replying to a Sender's offered QoS. This can be done automatically (simply replying with the Sender's offered QoS) or by inclusion of the user via the Graphical User interface. In the Receiver-initiated case the *Configurator* has to request the appropriate QoS for a flow in some way to the Sender.[81] If the QoS Service Provider does not supporting any kind of explicit end-to-end signalling - like a DiffServ Service Provider does - the sender can still start with sending information without any information about the client's capabilities. In the worst case the Receiver can not display the information received for a QoS aware flow in an appropriate manner.

Due the fact that the QoS Service Provider can be either Sender-initiated or Receiver-initiated considering a Sender QoS unaware Application behaves in the same way a Receiver QoS unaware Application does. So the third case is not considered explicitly. But it should be noted that on the sender's side it has to be guaranteed that the sender's socket (of the legacy Application) keeps the condition for their flows[82].

A5.12.3 How can a Configurator be supported

After analysing the task of a *Configurator* it has now to be discussed how it can take over the task of a QoS Proxy and gets access to the specific flows of the legacy Application. In general the *Configurator* can be supported either by the *Local Management Functionality* (see chapter A5.13) or by additional primitives offered by the *ESI* for supporting the *Configurator* in managing Legacy Applications. By the *Local Management Functionality* the Configuration can get access for example to the traffic control (packet classifier, admission control and packet scheduler) properties and can modify them through the *Local Management Interface*. The *ESI* can support the *Configurator* with new primitives, specific for legacy Applications or modifying the signature of already existing primitives. The latter is subject of this chapter and discussed in the following sections.

A5.12.4 Additional ESI primitives for supporting QoS unaware Applications

To support a *Legacy Application*'s flows with QoS the *Configurator* a) has to be informed about which flows it should take over the QoS management and b) it has to be notified about incoming 'QoS aware flow request' yet to be established and not handled by any application running on the mobile terminal. It is first considered the case 'a' where the *Configuration* is informed or configured due either a specific

[81] A way to do that might be offered by the means of QoSTemplates.

[82] A legacy Application has to be built with Shared Libs, so it is possible that the lib supporting the ESI can be linked during runtime.

configuration file or the user's input e.g. through the *Configurator* 's Graphical User Interface. The main issue on how a legacy application's flow can be addressed is explained in the following. There are different possibilities:

1. Explicitly specifying the flow by the means of a 5-tuple (*Source IP Address, Source Port, Protocol-ID, Remote IP Address, Remote Port*). How to get each element is a question directed to the operating system supported features.

2. Instead of addressing each parameter of the 5-tuple explicitly the usage of wildcards '*' can be introduced.

3. Specifying the flow with a 5-tuple (see 1) but instead of specifying a *Source IP Address* and *Source Port* the application's *Process-id* can be used.

To assign QoS to specific flows a flow descriptor is necessary (see definition of *ESI* primitives, chapter A5.8). The *ESI* has to support primitives allowing the specification of a flow in the above mentioned way and all matching flows has to be passed to the primitive's caller. Figure A5-24 depicts the way a *Configurator* can take over other's application QoS management for their flows. In a first step it has to register itself for the specific flows using the *ESI*'s *RegisterForFlow* primitive. After getting a set of flow descriptors matching the request the *Configurator* can associate QoS with the flow by using the common *ESI Services* like *SetQoS, AssignQoS, ChangeQoS* and *ReleaseQoS.* If the *Configurator* is not longer interested in managing QoS for a specific flow the *UnregisterForFlow* primitive can be triggered resulting in releasing QoS management for the specific flows.

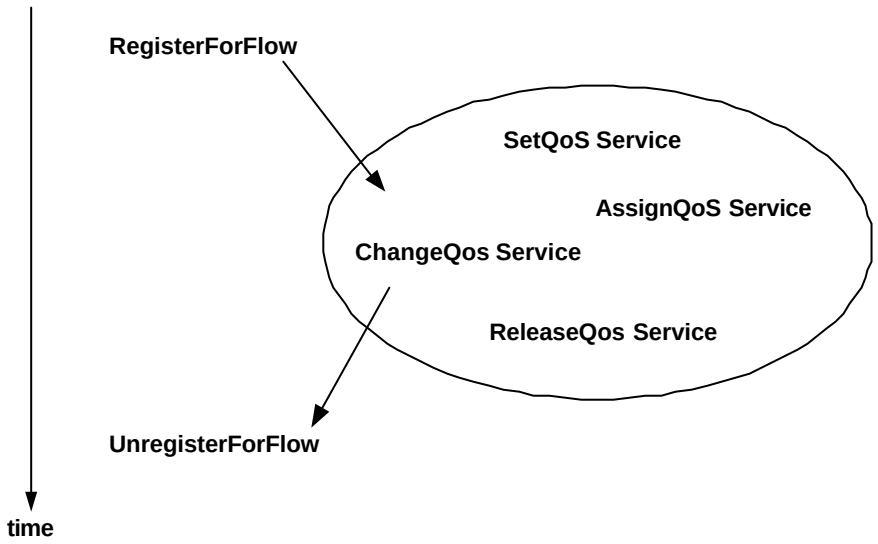


Figure A5-24: RegisterForFlow Service

As mentioned above, the information received after a *RegisterForFlow* primitive call is a set of flow descriptors used in *ESI Services*. This set consists of any flow matching the primitive's parameter. Note that this can be a lot of flows - and in some scenario the usage of wildcards or *Process-ID* is rather pointless. But nevertheless there are also meaningful scenarios where the usage of these parameters makes sense - and therefore they are provided by the *ESI.*

A5.12.4.1 RegisterForFlow and UnregisterForFlow Service

A detailed description of the additional introduced primitive *RegisterForFlow* and *UnregisterForFlow* are introduced in the following tables.

RegisterForFlow.request primitive	service user -> service provider

associated information	Description of flows the *Configurator* wants to manage QoS for. Can be specified in the following ways: Explicit specification of the 5-tuple: *Source IP Address, Source Port, Protocol-ID, Remote IP Address, Remote Port*. Using wildcards instead of specifying elements explicitly. Instead of *Source IP Address* and *Source Port* the legacy Application's *Process-id* can be used to specify the flow.
description	Specifies for which - already established flow - the *Configurator* takes over the QoS handling - so that the legacy Application gets QoS aware.

RegisterForFlow.confirm primitive	service provider -> service user
associated information	A set[83] of flow descriptors usable for associating the flow with QoS by using *ESI*'s *SetQoS, ChangeQoS* or *AssignQoS Services*.
description	The primitive call passes all flow descriptors matching the *RegisterForFlow.request* parameters, which are not already QoS aware.

UnregisterForFlow.request primitive	service user -> service provider
associated information	Description of flows the *Configurator* wants to release the managing of QoS for. Can be specified in the following ways: Explicit specification of the 5-tuple: *Source IP Address, Source Port, Protocol-ID, Remote IP Address, Remote Port*. Using wildcards instead of specifying elements explicitly. Instead of *Source IP Address* and *Source Port* the legacy Application's *Process-id* can be used to specify the flow.
description	Specifies for which flows the *Configurator* does not managing the QoS handling any longer.

Remarks:

It might be that *UnregisterForFlow.request* gives up the control for a QoS established flow without triggering the *ESI*'s *ReleaseQoS Service.* Either the Configurator involves the user with the decision or *ReleaseQoS* is accomplished implicitly for all flows matching the *UnregisterForFlow.request* parameters.

To use the *Process-ID* as parameter might be an optional case not supported by all mobile terminals operating system[84].

See section A5.12.4.3 how the primitives can be used.

A5.12.4.2 RegisterForQoSRequest and UnregisterForQoSRequest Service

The previous primitives are good if the *Configurator* plays the active role - like register itself for a specific flow. But there might be the case that the *Configurator* will be informed automatically about all incoming requests for establishing a QoS aware flow, which are not handled by any application. What does it mean - not handled by an Application? If a RSVP QoS Service Provider is considered, a application has to call the *Session.indication* primitive (see RFC [A5.6]) to announce its interested in participating RSVP handling. A Legacy Application can not do that - and therefore incoming request on the well-known port where no application feels responsible for can be forwarded to the *Configurator*, if it is interested in.

It is very useful for the *Configurator* on the Receiver side if a Sender-Initiated QoS Service Provider is used. In this case it can answer all incoming request for establishing a QoS aware flow by simply replying

[83] The kind of data structure passed with the primitive called is implementation specific.

[84] The author's personal view [OS].

with the Server's offered QoS. Note this is one way how the *Configurator* handle such kind of requests, an other way might be to involve the user in the decision process but that's open how the *Configurator* handles that request, the important thing is that there is a facility to do it automatically.

The *ESI* supports two primitives allowing the *Configurator* to register and unregister itself for all incoming requests for establishing a QoS aware flow. These are introduced in the following:

RegisterForQoSRequest.request primitive	service user -> service provider
associated information	none
description	A Configurator register itself at the *ESI* interested in getting all incoming requests relating to a QoS aware flow, not handled by any other application.

UnregisterForQoSRequest.request primitive	service user -> service provider
associated information	none
description	A Configurator unregister itself at the *ESI* resulting that all incoming requests relating to a QoS aware flow, not handled by any other application are handled in a predefined way - means that they might be ignored. (Depends on the basic behaviour of the QoS Service Provider responsible for the request).

See section A5.12.4.3 Usage Example how the primitives can be used.

A5.12.4.3 Usage Example

The following two figures illustrate the usage of the primitives introduced in the previous sections. Figure A5-24 shows a Receiver/Sender scenario assuming that the Receiver is a legacy application supported by the *Configurator*. It is further assumed that a logically connection is established meaning that a complete flow description is available on both sides. Note though the diagrams show the Receiver to the left of the *Configurator* does not mean that information from the *ESI* to the Receiver is passed through the *Configurator* - it is passed directly. After registering for the specific flow the *Configurator* manages the QoS for the specific flow by replying to the *SetQoS.indication* primitive with the *SetQoS.response* primitive passing the appropriate QoS Parameter (see chapter How the QoS Parameter are specified is out of scope of this example - rather a question of the functionality of the *Configurator*. After setting up the QoS aware flow data can be transmitted from the Sender to the Receiver on a QoS aware flow. At a specific point - triggered for example by the user - the *Configurator* unregister itself for the specific flow - after releasing all request resources due *ReleaseQoS*.

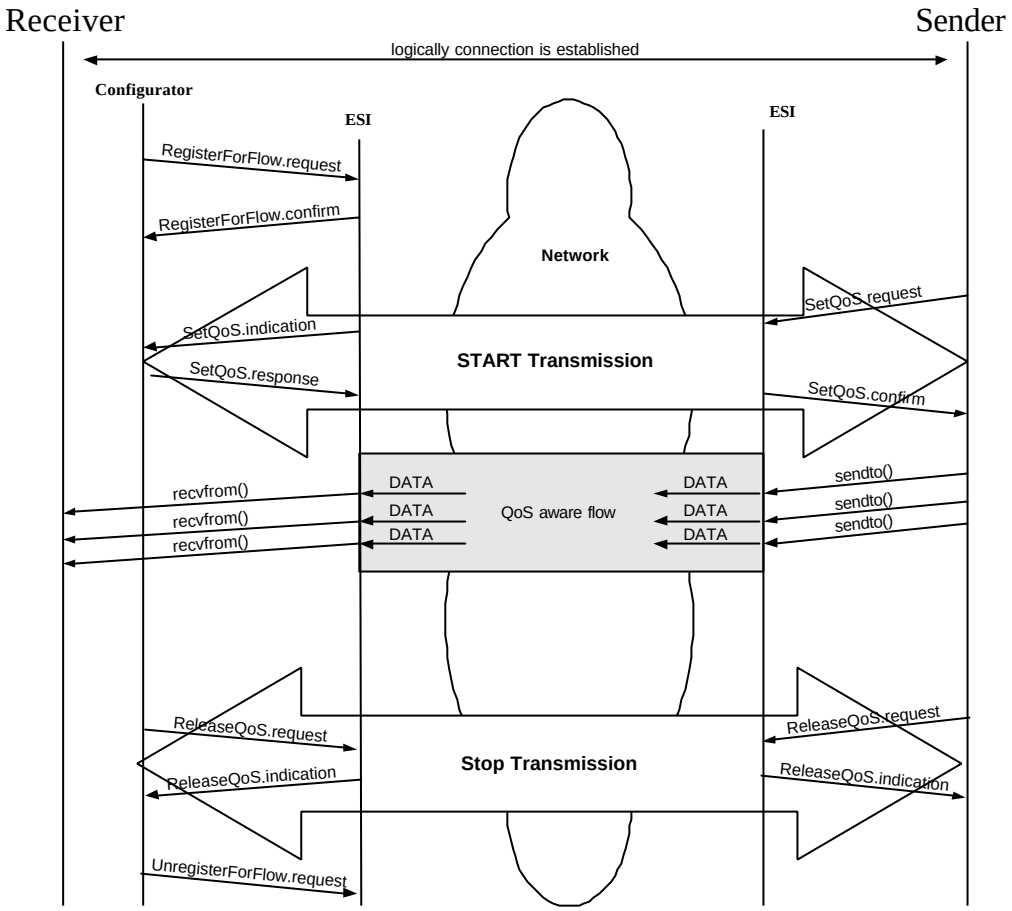


Figure A5-25: Usage of RegisterForFlow and UnregisterForFlow

Figure A5-26 shows the usage of the *RegisterForQoSRequest.indication* and *UnregisterForQoSRequest.indication* from a Receiver point of view. Note though the diagrams show the Receiver to the left of the *Configurator* does not mean that information from the *ESI* to the Receiver is passed through the *Configurator* - it is passed directly. In the example a RSVP QoS Service Provider is assume for the sake of explanation. The Configurator show its interest in managing QoS for flows by simply calling the *RegisterForQoSRequest.indication* primitive resulting in forwarding all request for QoS aware flows to the *Configurator.* Incoming *Sender.indication* primitives, mapped to a *SetQoS.indication* primitive are forwarded to the *Configurator* which is henceforth responsible for handling the QoS for that flow. At least the *Configurator* gets the *Sender.indication*. If it is replying depends on how the *Configurator* acts in detail. If the *Configurator* is not longer interesting the *UnregisterForQoSRequest.request* result in that the *ESI* handles request to QoS aware flow in the default way - depends on the default behaviour of the usable QoS Service Provider.

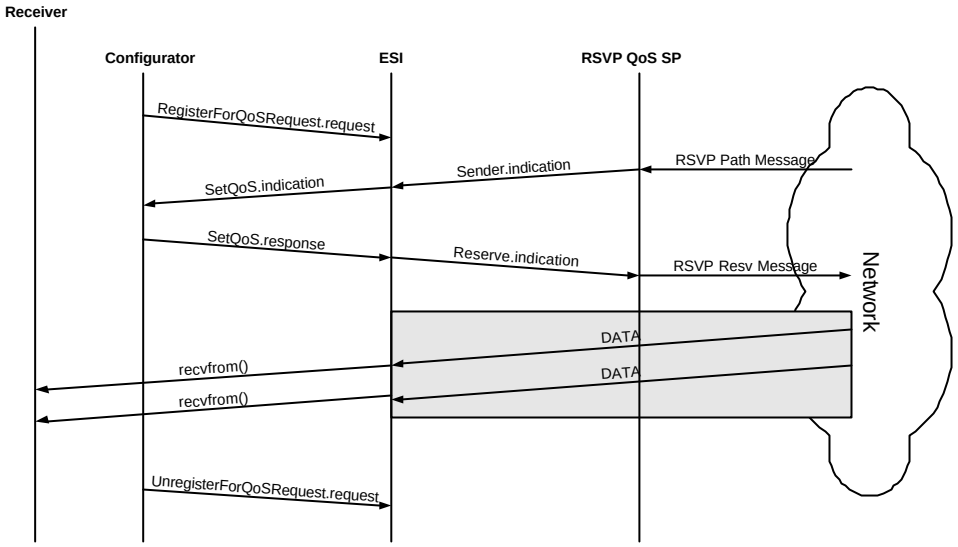


Figure A5-26: Register / UnregisterForQoSRequest

A5.13 Local Management Functionality

As depicted in Figure 5-1: an additional interface is defined beside the *ESI*. It aims to support multimedia applications with different management functions. The detail about the interface and the it's functionality is introduced in the following sections.

A5.13.1 Background

Future broadband wireless multimedia applications shall be able to run on a set of different terminals. These terminals might support a variety of different networks. In order to be aware of the terminal capabilities and to manipulate the operation of the terminal, applications usually interact with the operating system to discover available network adapters, the state of the network and other required features. Furthermore, the operating system usually provides control functions that allow fine grain control over the behaviour of the terminal.

A set of mandatory and optional management functions has been identified that are useful to support specialised multimedia applications. In order to avoid operating system specific functions, local management functions in terms of an abstract object model are defined. The advantages of this approach are two-folded. First, it allows to concentrate on the management functions itself, without being distracted by the way, a specific operating system might implement this. Second, it allows mapping this specification to an API that abstracts from OS specific functions. This layer between the OS and the applications enhances the portability of applications.

The following sections are structured like this: First, we describe the abstract object model that allows specifying the required functionality. Then an overview about the management function is identified.

A5.13.2 Management Model

The Management Model describes management functions through a standard object model with an event mechanism. Furthermore, the management provides a mechanism to detect which management objects are available.

Management Objects
A Management Object represents a single functionality inside of a BRAIN protocol stack. Management objects consist of:

- **Attributes** that describe the state of the object. For simplicity reasons, attributes can only be basic data types like integers or strings.
- **Methods** that allows invoking operations on the object.
- A set of **child** objects that allows representing dynamic aspects of the system.
- A **name** that identifies an object uniquely in the set of siblings (e.g. /adapter/ethernet1, /adapter/ethernet2, /adapter/hiperlan1).

Management Events
Interested applications can register for management events that can be issued from an object. There are two types of events:

- The first event signals the change of an attribute of a management object. It returns the name of the changed attribute as well as the new value.
- The second event signals the change in the set of children. It returns the name of the affected child as well as this object was inserted into the child set or removed.

Discovery Functions
In order to allow the flexible usage of the event model, each terminal is allowed to implement different sets of management object. This gives additional flexibility to the terminal manufacturer. In order to cope with this, applications need a way to discover available management objects. In order to do so, the system provides a root object named "/" that represents the logical collections of available management objects. A predefined method listChildren() returns the names of the children objects. The application can then access these objects and apply the listChildren() function recursively.

Remarks:
The main purpose of this model is to provide an abstract object model to describe management functionality. Therefore, the model presented here tries to provide a minimum set of required functionality. A concrete implementation would probably add functions that allow a sophisticated and efficient usage of the model. For example, instead of registering with each object for events, a concrete

implementation might offer a function that automatically delivers all events from a single sub-tree of the management functions.

A5.13.3 BRAIN Management Functions

The following section describes several identified local management functions. This is by far not the complete set of required local management functions, but deal with certain aspects of the system.

A5.13.3.1 Network Adapters

BRAIN applications want to identify the available network adapter and their status. The system provides a management object called "/networkadapter" that monitors the available network adapters. For each adapter it provides a child object that represents the adapter and its status attributes. (see also section A5.15.3.2 Network Interface Card related parameter (NIC))

Name	Attribute/Method	Description
/networkadapter		MO representing a list of available Network Provider:
/networkadapter/eth0		One example network adapter. There is NO naming convention on the name of the adapter.
	Type	Type of network, e.g. "Ethernet", "HIPERLAN/2"
	Status	0 = down 1 = up
	MaxBandwidth	Maximum available bandwidth on this adapter, e.g. 10Mbit/s on an standard Ethernet adapter
	AddMonitor()	Adds a monitoring child object
	RemoveMonitor()	There can be only one.
/networkadapter/eth0/ monitor		A object monitoring the current network adapter
	UsedBandwidth_sec	The bandwidth used in the last second
	UsedBandwidth_min	The bandwidth used in the last minute
	UsedBandwidth_h	The bandwidth used in the last hour.

A5.13.3.2 Enhanced Service Layer

According to WP1, BRAIN applications might want to select the QoS SP used by the Enhanced Socket Layer. Whenever a terminal manufacturer provides this capability, the following set of management object shall be used to give applications the required control.

Name	Attribute/Method	Description
/ESL		Management Object representing the *ESL*
/ESL/QoSSPs		MO representing a list of available QoS SP:
	StandardQoSSP	The currently active standard QoS (at start-up time, this is pre-configured, can be changed with the SelectQoSSP function).
	SelectQoSSP()	Method to set the new QoS SP.
/ESL/QoSSPs/RSVP_QOSSP		A RSVP QoS SP.
	Type	Type of the QoS SP

A5.13.3.3 Mobility functionality

Local Management Interface offers functionality that enables the user to monitor handover functionality. It provides mechanisms to inform about ongoing handovers.

Name	Attribute/Method	Description
/MobilityManagement		Contains status information about the Mobility MO. Is used to be informed about possible handover opportunities.
/MobilityManagement/Handover		MO representing a list of supported

		handover types.
	addMonitor()	adds a monitor object as child
	removeMonitor()	removes a running monitor object
/MobilityManagement/Handover/ Monitor		
	Status	0 = no ongoing handover 1 = planned handover 2 = forced / unplanned handover

Based on this information an application can accomplish specific tasks.

A5.14 Analyse Mobility Management-related aspects of the ESI

At the beginning of IP based networks, the end systems –known as hosts- had a single physical point of attachment (network interface) to the network. This network interface was identified by an IP address. This address had to be known to establish a connection with a remote host, thus this IP address somehow served as identification of the host. When different services were run at the same host, an additional number had to be supplied to identify each different service. It was termed port number. Lately the need to run different instances of the same service in a host appeared. As an example, consider a host attending requests to two different web servers (e.g. www.info.net and www.news.org). The port (service identifier) is the same while different web pages should be served. To solve these situations, the IP aliasing concept was introduced, allowing a host to have several IP addresses associated with the same network interface. At this time, the IP address could no longer be used as the unique host identifier.

Another evolution of the single network interface host was the multi-homed host. This is a host with several network interfaces, thus being able to send/receive traffic to/from different IP networks. The main difference with a router is that it acts as source or sink for the traffic flows. This type of hosts required special support from its operating system, mainly in two aspects. Firstly, the routing table needed to store an additional parameter, the outgoing network interface to reach the next hop. Secondly when a connection is established the application had to have control on the interface used. The standard socket interface allowed this possibility. The application had to provide the required source address in addition to the destination address. This information has to be used by the local operating system to assign the connection to the network interface configured, which such address. The application obtained the local IP addresses queering to the operating system.

The BRAIN usage scenarios [A5.2] introduced some situations where the terminal must be aware of an additional information: the network provider (NP). The NP is an entity offering IP based communication services. With the introduction of radio based network interfaces –cellular systems or wireless LANs- the terminal faces a new situation: several NPs offering simultaneously IP based communications at a given location. Then the terminal must somehow choose one of them to establish network connections. Several reasons can influence the selection of a NP when establishing a connection. The first one would be that the NP can reach the intended destination. Recall that BRAIN usage scenarios consider the possibility of a NP which deploys a wireless network offering access to a limited set of services (e.g. the wireless network of an airport). This NP cannot be used to establish Internet scoped connections, while it must be used to access the airport local information. The discovery of the reachable destinations through a NP does not need new developments, current IP protocols already support this functionality. The terminal can use ICMP messages to find out if a particular destination is reachable through a NP at any time. In addition the NP could broadcast routing information to the terminals attached to its network.

In some situations, several NPs will be able to deliver traffic to a destination. Then the terminal must be able to acquire extra information to choose one of them. Typically the terminal could check on the additional features to the basic IP service. For example it could check if there are enough resources at the NP to provide the required QoS for the connection; or what are the security mechanisms in place for the transmission (authentication or privacy); or if there are special servers deployed at the NP to easy the transmission (e.g. a local email relay). All this type of information can be obtained using current protocols, what is new is the need for the terminal to gather this information and store it. This information could then be used by the module of the ESL automatically assigning new connections to NPs or by applications selecting the NP for their connections through the Local Management Interface.

This last sentence raises the question on which entity should select the NP for a connection. Two modes are supported: automatic and manual NP selection. Most of the applications will use the automatic mode, that is relying in the terminal to choose the best NP for each connection. The entity in the ESL that will perform this function automatically is the Primitive Mapper. For the decision it will consider the information gathered from the available NPs, the parameters supplied by the application in the connection request (e.g. intended destination, QoS desired) and the user profiles and system policies stored in the terminal. Some applications might prefer to control the NP used for their connections. This is the manual NP selection mode supported through a set of primitives in the Local Management Interface (see chapter A5.13).

?? **EnumerateNWProvider([capabilities])**

The EnumerateNWProvider primitive is used to retrieve information about available network providers. The optional capabilities parameter could be used to provide a profile of the specific network provider capabilities required (e.g. mobility support/without mobility support). The

primitive returns a list of Network Providers that are able to fulfil the indicated capabilities. Each NP is identified by a code and a text string. The text string will be a name familiar to the user, while the code will be used internally by the terminal entities. As an example, EnumerateNWProvider() could return {(134,Vodafone) (2345,Mall_NP)}, while EnumerateNWProvider(mobility support) could return {(134,Vodafone)}. The enumeration of the complete set of capabilities offered by NPs is outside the scope of this document, as this will be primarily market driven.

?? **GetInfoNWProvider(NWProvider_code)**

The GetInfoNWProvider retrieves information about the operational characteristics and performance of a given network provider. As a NP can be accessed from a terminal using different network interfaces (e.g. a NP which offers UMTS and GSM services), the answer will be classified by the network interfaces of the terminal. As an example, GetInfoNWPRovider(2345) could return {/networkadaptor/UMTS (capabilities_list, performance_param), /networkadaptor/GSM (capabilities_list, performance_param)}. The network interfaces correspond to these retrieved through the Local Management Interface. The capabilities list corresponds to the one commented in the previous primitive. The performance parameters are described in similar terms than the QoS. For more details of information, see section A5.15.3.2 Network Interface Card related parameter (NIC).

?? **SelectNWProvider(flow, NWProvider_code, network_interface)**

The SelectNWProvider primitive sets the network provider and the network interface for the specified flow. To perform this assignment, the application is expected to have its internal policy. As an example, an application-based policy could look like: first, use network provider for a fixed network interface; if none is available, use 802.11 conform wireless NP due to cost constraints; finally, use UMTS NP if no other available. This policy will be applied to assign each flow to a pair (NP, network interface).

To improve the performance when creating a new flow, an application typically will use the primitives EnumerateNWProvider and GetInfoNWProvider in advance, storing the retrieved information. In this way, the information is ready at the time the flow is created. To maintain this information updated, applications should poll regularly the network, even if no new flows are expected. In order to avoid this inefficiency, the Local Management Interface allows applications to register for certain events related to NPs behaviour. These are the indications that applications can register for:

?? **NWProviderReachable.Indication(NWProvider_code_list)**

Issued when a new network provider becomes reachable or a previously reachable NP is now unreachable. As parameter the list with the codes of the affected NPs is included.

?? **NWProviderForcedHandOver.Indication(NWProvider_code, old_NI, new_NI)**

Issued when a NWProvider performed a forced vertical handover (e.g. from HIPERLAN/2 to UMTS). If the handover is horizontal, the application is not informed with this indication as no mayor disruption of the service capabilities are expected. The level of QoS could be affected due to the lack of resources in the new access point, but this is notified as a QoS violation. As parameters it contains the code of the affected NP, the old network interface and the new network interface.

?? **NWProviderHandOverPlanned.Indication(NWProvider_code, current_NI, future_NI)**

Issued when a vertical handover is planned in the near future. As parameter it contains the code of the NP, the current network interface and the future one. Based on this information and the information about alternative network interfaces for the NP, the application can proactively prepare itself for the handover –e.g. increasing the buffering- or specify a different network interface for the vertical handover using **SelectNWProvider()** .

A5.15 Appendix

This chapter contains the background knowledge rose up during different discussion in WP2 and terminology of used terms in this document.

A5.15.1 Terminology

flow An individual, uni-directional data stream between two transport layer entities, uniquely identified by a flow identifier like a 5-tuple containing source address, source port, protocol type, remote address, remote port. Note that a flow is supposed to exist for a longer period of time. A flow can be described as data packets traveling hop-by-hop through the network from the originating transport entity on a specific host to a receiving transport entity at the destination host or hosts.

A5.15.2 Explicit Congestion Notification

This chapter contains the issue of how *Explicit Congestion Notification* can be handled to support the application (if requested) with information about it. Detailed information can be found in [A5.4]**.**

QoS related issue (see [A5.4]**)**
If network resources are running out, various queues are growing and routers are dropping packets. Both the applications and the network would benefit from explicit indications of these problems: applications can take measures to lower their transmission speed or a VoIP application can start to use a more robust coding scheme; the lowered amount of data transferred enables the network to clear the congestion situation. Even without explicit notifications, some transfer protocols watch their transfer and react to probable congestion. If packets are dropped or sufficiently delayed, for example, TCP halves its sending rate and RTP can try to switch between multimedia codecs.

Explicit Congestion Notification (ECN, [A5.17]) is becoming a central element in reporting of congestion in a network. ECN is so far studied to be used with TCP, and work is ongoing towards using ECN with UDP-based flows. Therefore, we suggest using the ECN framework to notify of network congestion in the BRAIN network. In addition, the IP2W interface provides a mechanism to provide feedback to MNs on problems related specifically to the wireless link.

ESI/LMI related issue
The feedback about ECN can be passed to the application layer either through the *Local Management Functionality* or it can be mapped to a *QoSViolation.indication* primitive (if specific QoS is violated) with the parameter *type* value's ECN. This has to be specified into more detail, it might be that the type can be expressed in a more generic way, nevertheless the information - if available - should be passed to the application.

A5.15.3 Requests from Work Group 1

This chapter contains requests from Work Group 1, which have to be considered in further design and implementation phases.

A5.15.3.1 Local Management Functionality related

WP1 requested to have the possibility to set a network provider explicitly for a specific flow - a primitive called SelectNWProvider(flow, NWProvider). This method is especially useful in a multi-homed terminal (see chapter A5.8.9.1). As an example, the application might always use the 802.11 conform wireless network provider (if available) due to cost constraints. However, the application would use as its primer network provider the gigabit Ethernet conform NWProvider due to performance constraints. Therefore an application-based policy could look like - first, use network provider for a fixed network. If none available, use 802.11 conform wireless NWProvider due to cost constraints. Finally, use UMTS NWProvider if no other available.

A5.15.3.2 Network Interface Card related parameter (NIC)

Note, the content of this section is exact the same as send out with the WP-SM030-1b-PI document.

This section describes parameters that are of interest for WP1. These parameters allow the QoS framework to retrieve operational and performance-related information in order to make policy driven QoS decisions. This allows making adaptation cycles in advance. Note that this describes parameters that should be available through the Local Management Interface. Note, in a multi-homed terminal, there

might be more NIC (Network Interface Cards) running in parallel. There should be a mechanism to address each NIC, separately for gathering per NIC statistics.

To be in line with the Local Management Functionality (introduced in chapter A5.13) the following characteristics can be implemented as attributes of the Network Adapter Object for example for the first HIPERLAN Interface : /networkadapter/hl0)

General Operational Characteristics

Name	Description
GEN_SUPPORTED_LIST	List of Supported Parameters
GEN_HARDWARE_STATUS	Specifies the current hardware status of the underlying NIC: Parameters could be one of the following: Ready Initializing Closing Reset Not Ready
GEN_MEDIA_SUPPORTED	Media types supported. Parameters could be one or a list of the following: MEDIUM_HIPERLAN/2 MEDIUM_802.11 MEDIUM_UMTS ...
GEN_MEDIA_IN_USE	Specifies a complete list of the media types that the NIC currently uses. This list can include some, none , or all of the above.
GEN_QOS_SUPPORTED	Specifies if and to what extend QoS is supported. Parameters could be one of the following: QOS_BEST_EFFORT QOS_CONTROLLED_LOAD QOS_PREDICTABLE_QOS ....
GEN_MAXIMUM_FRAME_SIZE	Specifies the maximum network packet size, in bytes, that the NIC supports. In response to this query from requesting transports, the NIC driver should indicate the maximum frame size that the transports can send, excluding the header. This parameter can be used by WP1 to determine the packet size of the PDUs for compressed multimedia data in order to optimise throughput.
GEN_LINK_SPEED	Specifies the maximum speed of the NIC in kbps. This parameter can be used in WP1 for the broker to distribute the possible link capacity among all QoS managed connections when performing QoS orchestration.
GEN_VENDOR_ID	Specifies a three-byte IEEE-registered vendor code, followed by a single byte that the vendor assigns to identify a particular NIC. The IEEE code uniquely identifies the vendor and is the same as the three bytes appearing at the beginning of the NIC hardware address. In our case this code could be HI-2
GEN_VENDOR_DESCRIPTION	Pints to a zero-terminated, counted string describing the NIC
GEN_MEDIA_CONNECT_STATUS	Returns the connection status of the NIC on the network as one of the following values: **MediaStateConnected**

	M**diaStateDisconnected**
GEN_PHYSICAL_MEDIUM	Specifies the types of physical media that the NIC supports. This parameter is an extension of GEN_MEDIA_SUPPORTED. NICs use this Parameter to differentiate their physical media from media that they declared to support in the GEN_MEDIA_SUPPORTED query. These media types could be one of: **PhysicalMediumWirelessLan** Packets are transferred over a wireless LAN network. Includes, for example, IEEE 802.11

General Statistics

The following attributes can be provided by a network monitors (see A5.13.3.1) for all network adapters, e.g. /networkadapter/eth0/monitor.

These Parameters describe the general statistics. With frames, we denote the units used by the NIC, i.e. MAC-frames. Note that these statistics are applicable from the start of the system up to where the statistics are gathered. As an example, directly after start, the value of GEN_XMIT_OK is 0, after 1 sec it could be 300, after 2 seconds 450,...

Name	Description
GEN_XMIT_OK	specifies the number of frames that are transmitted without errors.
GEN_RCV_OK	specifies the number of frames that the NIC receives without errors.
GEN_XMIT_ERROR	specifies the number of frames that a NIC fails to transmit
GEN_XMIT_BUFFER_ERROR	specifies the number of frames that a NIC fails to transmit due to limited buffer space. This helps applications to decide to slow down the rate of transmission
GEN_RCV_ERROR	specifies the number of frames that a NIC receives but does not indicate to the protocols due to errors.
GEN_DIRECTED_BYTES_XMIT	specifies the number of bytes in directed packets that are transmitted without errors.
GEN_MULTICAST_BYTES_XMIT	specifies the number of bytes in multicast/functional packets that are transmitted without errors
GEN_BROADCAST_BYTES_XMIT	specifies the number of bytes in broadcast packets that are transmitted without errors
GEN_DIRECTED_BYTES_RCV	specifies the number of bytes in directed packets that are received without errors
GEN_MULTICAST_BYTES_RCV	specifies the number of bytes in multicast/functional packets that are received without errors
GEN_BROADCAST_BYTES_RCV	specifies the number of bytes in broadcast packets that are received without errors
GEN_RCV_CRC_ERROR	specifies the number of frames that are received with checksum errors
GEN_RCV_BUFFER_ERROR	specifies the number of frames that a NIC fails to deliver to higher layers due to limited buffer space. This helps applications to decide to start downgrade adaptation.
GEN_TRANSMIT_QUEUE_LENGTH	specifies the number of packets that are currently queued for transmission. For queries, the number returned is always the total number of packets currently queued.

Wireless Operational Characteristics

The following attributes can be provided by a network monitors (See A5.13.3.1) for wireless network adapters, e.g. /networkadapter/hl0/monitor.

These Parameters describe specific wireless operational characteristics. NIC denotes network interface card and stands for one radio transceiver unit (e.g. HIPERLAN/2 or UMTS). Of course, if the radio transceiver unit supports several technologies simultaneously, these statistics apply per technology.

Name	Description
WW_GEN_OPERATION_MODE	return information about the NIC's current power saving mode as: **normal** **powersaving_on**
WW_GEN_DISABLE_TRANSMITTER	return the NIC's current transmitter status as: enabled disabled
WW_GEN_NETWORK_ID	return the ID of the network with which its NIC is currently configured to communicate
WW_GEN_BASESTATION_ID	return the ID of the base station or adhoc device last contacted by the NIC
WW_GEN_ENCRYPTION_SUPPORTED	return the type(s) of encryption supported by the NIC. The following lists the valid encryption types **WWUnknownEncryption** **WWNoEncryption** No support for encryption is available. **WWDESEncryption** The NIC supports DES encryption. **WWRC2Encryption** The NIC supports RC2 encryption. **WWRC4Encryption** The NIC supports RC4 encryption. **WWRC5Encryption** The NIC supports RC5 encryption.
WW_GEN_ENCRYPTION_IN_USE	returns the type of encryption currently in use
WW_GEN_CHANNEL_QUALITY	return information about the quality of the link between its NIC and the network as follows 0 The wireless NIC is not in contact with the network. 1-100 The NIC can communicate with the base station at the given quality for the link, expressed as a normalized value. 100 designate the highest possible quality for the link. -1 Channel quality is unknown.
WW_GEN_REGISTRATION_STATUS	return the current registration state of its NIC on the network as follows 0 Registration was denied. 1 Registration is pending on the network. 2 Registered on the network.

	-1 Registration status is unknown.
WW_GEN_RADIO_LINK_SPEED_TX	returns the current radio link speed in bps for transmitting
WW_GEN_RADIO_LINK_SPEED_RX	returns the current radio link speed in bps for receiving
WW_GEN_LATENCY	returns current estimate of the minimum latency, in milliseconds, for sending a net packet of the maximum size permitted by the network from one end point to the other within the BAN.
WW_GEN_RSSI	returns information that specifies the strength of the signal in decibels (dBm) that the NIC receives. This parameter requests the raw signal strength versus the normalised value.

A5.16 Enhanced Service Interface References

[A5.1] [Georg Neureiter, Louise Burness, Andreas Kassler, Piyush Khengar, Ernö Kovacs, Davide Mandato,Jukka Manner, Tomàs Robles, Hector Velayos: “The BRAIN Quality of Service Architecture forAdaptable Services with Mobility Support”, to appear in PIMRC 2000, Sept. 2000, London, England

[A5.2] Scenarios for mobile IP services and resulting requirements in different wireless networks, deliverable working group 1

[A5.3] Concepts for Service adaptation, scalability and QoS handling on mobility enabled networks.Deliverable Working Group one.

[A5.4] WP2 Document worked out by the QoS Working Group - part of [D2.2]/ QoS Section

[A5.5] L.L. Peterson, B.S. Davie: “Computer Networks, A systems approach”, Morgan Kaufmann Pub. 1996, p. 458

[A5.6] Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification

[A5.7] The Use of RSVP with IETF Integrated Services - J. Wroclawski, MIT LCS, September 1997

[A5.8] Andrew S.Tanenbaum, Computer Networks - THIRD Edition, Prentice HALL Pub1996 p 810

[A5.9] Unix Network Programming, W.Richard Stevens, Prentice Hall Software Series

[A5.10] WP1-SO028, Andreas Kassler and Davide Mandato, "End-to-End QoS Negotiation Protocol"

[A5.11] P. Pan, and H. Schulzrinne, "YESSIR: A Simple Reservation Mechanism for the Internet", Computer Communication Review, Vol. 29, No. 2, April 1999. (This is a minor changed version of the NOSSDAV'98 paper)

[A5.12] Internet Stream Protocol Version 2 (ST2) Protocol Specification -Version ST2+, RFC 1819

[A5.13] Bernet, Y., Yavatkar, R., Ford, P., Baker, F., Zhang, L., Speer, M., Braden, R., Davie, R., Wroclawski, J., Felstaine, E., "A Framework For Integrated Services Operation Over Diffserv Networks". Internet Engineering Task Force, Internet Draft, draft-ietf-issll-diffserv-rsvp-05.txt, May, 2000 (expires November, 2000).

[A5.14] S.Blake, D.Black, M.Carlson, E. Davies, Z. Wang, W. Weiss " An Architecture for Differentiated Services", Internet Engineering Task Force, RFC 2475

[A5.15] The Open Book, A Practical Perspective on OSI, Marshall T. Rose Prentice Hall, Englewood Cliffs, N.J. 07632

[A5.16] ITU E.800 Quality of service and dependability vocabulary 1994

[A5.17] Ramakrishnan, K., Floyd, S., "A Proposal to add Explicit Congestion Notification (ECN) to IP". Internet Engineering Task Force, Request for Comments (Status: Experimental) 2481, January 1999.

A6 IP2W Interface Annex

A6.1 IP$_2$W Convergence Model

This Annex presents the IP to Wireless convergence (IP$_2$W) model and the interface specification. IP$_2$W defines interfaces for controlling specific link layer features. It identifies a set of functions and functional requirements as well as a set of recommendations on how to design wireless link layers in a way that facilitates IP mobility, IP QoS, and efficient transmission of IP traffic in general.

A6.1.1 Overview

Figure A6-1 presents the layers of the TCP/IP stack, link-layer and the relation of the IP$_2$W interface to these.

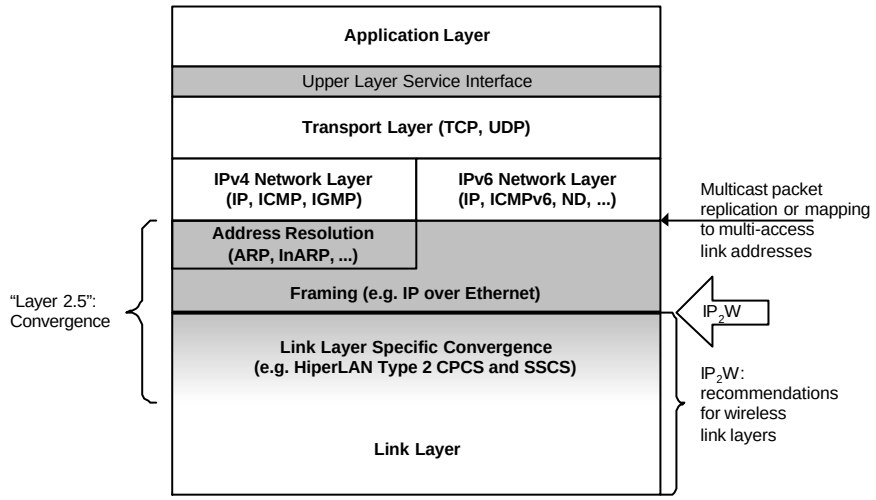


Figure A6-1: TCP/IP Protocols and IP$_2$W Interface

Motivation for the interface stems from the characteristics of wireless links; they are very different compared to fixed links. This poses special requirements on the interworking between the network layer and the wireless link layer. Radio resources are typically scarce and packet loss may be extensive. The point of attachment to an access network may change suddenly, which inflicts fluctuations in QoS and may cause a need to change the routing path. Moreover, it has been widely recognised that assistance from link layer mechanisms is prerequisite for devising efficient fast handover solutions for wireless IP access networks. These issues call for designing a uniform wireless-enhanced interface for transmitting IP packets over wireless links.

There has been little previous work dedicated to this task. To some extent, a subset of this functionality can be found in any existing implementation. However it has been geared towards supporting the IP layer in a minimal fashion. Even if there was some provision for wireless traffic, the interfaces have been ad-hoc in nature and tied to a particular link layer technology.

To avoid similar problems, a set of design goals have been applied in designing the IP$_2$W interface:

?? Provide a unified interface for controlling the various capabilities of wireless interfaces, including support for QoS, resource reservations, efficient handover and idle mode.

?? Make useful information from the link layer available to upper layers, including application layer, e.g., in order to assists in handover and address acquisition.

?? Provide a platform for supporting (at least) mobile controlled handover between different air interfaces (vertical handover).

?? Allow for link layer specific optimisations of various IP specific features in a manner that is transparent to upper layers.

?? Do not attempt to make policy decisions within the link layer or in the interface design.

?? Do not compromise layer transparency. Define the added functionality in the form of a service interface and inter-layer hints.

The IP_2W interface is presented in Figure A6-2. It is separated into a data interface and a control interface (i.e., a separation between control plane and user plane is identified). Each interface offers access to a set of functionality on the link layer. Several distinct functions have been identified under the interfaces, represented by the small ovals. The ovals surrounded by broken lines represent optional functions, which may or may not be supported by the wireless link. If supported, however, the requirements and recommendations given in the IP_2W specification should be applied.

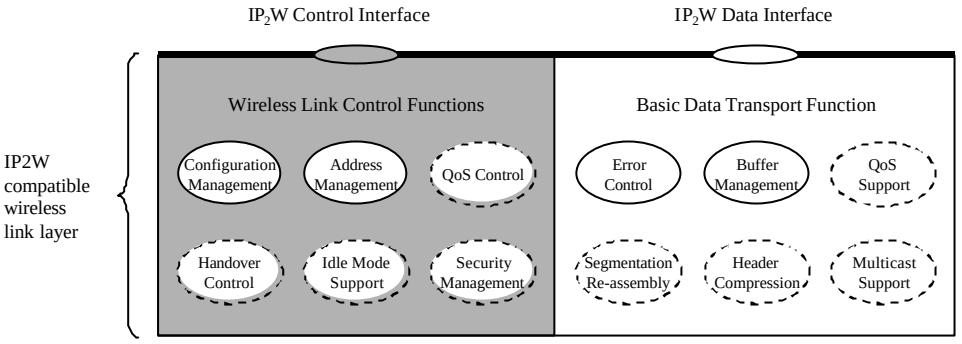


Figure A6-2: IP2W Convergence Interface

The functional blocks below the IP_2W Control Interface indicate functions that can be configured and controlled through the interface. These include Configuration Management, interface for querying the capabilities of the link layer, Address Management, Quality-of-Service (QoS) Control, Handover Control, Idle Mode Support and Security Management.

Data Interface consists of Error Control, mechanisms used to detect and correct errors on the link layer. Buffer Management refers to how buffers are managed on link layer with regards to congestion, flow control and other issues. QoS Support schedules packets to radio link channels. Segmentation and Re-assembly refer to supporting links, which do not support minimum MTU sized packets. Header Compression can optionally be performed in the link-layer and Multicast Support refers to supporting native multicast transmission.

The set of functions is not intended to be a complete one, but rather identifies the functions that are of interest to upper layers. The model does not attempt to mandate any specific organisation of functionality inside a particular link-layer technology. Nor does it imply any particular way of dividing the functionality into sub-layers inside the link-layer. The same degrees of freedom apply to the functional blocks in the IP_2W Data Interface. These functions have been identified in order to organise the requirements and recommendations on user plane procedures in a coherent way. The functions have no direct implication on the IP_2W data interface. However, the behaviour of some of the user plane procedures can be adjusted through the control interface.

The IP_2W interface aims to be generic enough to be applicable to different wireless link layer technologies, yet detailed enough to preclude the need at upper layers to utilise any functionality or information that is specific to a particular wireless technology. This flexibility is achieved by dividing the IP_2W functional blocks into specific capabilities, some of which are considered optional. An IP_2W compliant link layer advertises the capabilities it supports through a configuration function, allowing the higher layers to adjust to the characteristics and capabilities of the link layer.

IP_2W is only one piece of the whole. It is always coupled with an implementation of a “convergence layer” for a specific link layer technology. In the BRAIN architecture the link-layers used in Mobile Node (MN) and the Brain Access Router (BAR) are expected to comply with this interface. Both MN and BAR rely on IP_2W capabilities to support fast handovers at the IP layer and to provide efficient IP packet transfer and QoS reservations over the wireless link. The IP_2W capabilities in turn are based on the more elementary services that are provided by the wireless link layer.

This section is organised into sub-sections according to the functional blocks as defined above. Section A6.1.2 describes the Address Management issues; section A6.1.3 discusses requirements and solutions for QoS Control; section A6.1.3 describes the Handover Control functions; section A6.1.5 addresses the Idle Mode support; section A6.1.6 discusses Security issues; section A6.1.7 summarises the required Configuration information; section A6.1.8 discusses the requirements imposed on the data interface by all of these functions. Finally, section A6.2 contains the interface specification.

A6.1.2 Address Management

Address Management includes the allocation and managing of link layer and network layer addresses, and mapping between these addresses. IP_2W Address Management interface should cope with a wide variety of different links and allow for an efficient implementation of IP specific functionality, nevertheless without excluding non-IP based network layers.

The under-lying link-layer should implement at least the minimum functionality required for supporting the IP version 4 and 6. This might mean finding ways to support functionality not provided by the link natively, such as multicasting. The interface should be generic enough that advanced link-layers can export extended functionality in a sensible manner. The network layer can then use this extended functionality to implement its functions in an optimised manner. For example, special care should be taken to enable fast address acquisition in case of handover type situations.

A6.1.2.1 IPv6 Address Management

The operation of the IPv6 address management mechanisms and their requirements to the lower layers are discussed in this section. Of these mechanisms, address acquisition may be needed during handoffs. Then its performance plays integral role to the overall handoff performance.

Mobile Nodes conforming to Mobile IPv6 [A6.1] perform movement detection to detect change of location from one link to another. Mobile IPv6 defines a scheme for movement detection using the facilities of the IPv6 Neighbor Discovery [A6.2] such as Router Discovery and Neighbor Unreachability Detection. These mechanisms are designed to work independently of the link layer to enable implementations work in a variety of environments. However if link layer can provide additional information for possibly faster and more robust movement detection, it can be used by specific implementations.

Each IPv6 host has multiple IP addresses that the host is required to identify to it self [A6.3]:

?? Link-local addresses of each interface are used when communicating with network elements on the same link as the host.

?? Assigned unicast addresses of site or global scope are used when communicating with other hosts within the same site or the Internet.

?? Loopback address is a host internal address.

?? All-nodes multicast address is a link-local multicast address used for Router Advertisements and when it is required to reach all nodes on the link.

?? Solicited-node multicast address; One for each of its assigned unicast and anycast addresses. They are formed by taking the low-order 24 bits of the address and appending those bits to the prefix FF02:0:0:0:0:1:FF00::/104. They are used for Neighbor Discovery messages.

?? Multicast addresses of all other groups to which the node belongs.

In addition to that routers have to recognise the following addresses:

?? The Subnet-router anycast addresses for the interfaces it is configured to act as a router on.

?? All other anycast addresses with which the router has been configured

?? All-routers multicast addresses that are used for Router Solicitations and when reaching all routers on the link.

?? Multicast addresses of all other groups to which the router belongs.

There must be a link-layer address that corresponds to each of these addresses. The relation is a not one to one mapping because usually interface has only one unicast link-layer address it recognises. Network layer uses link-layer addresses to identify interfaces on the link. Link-layer is not required to use the same address as its hardware link address although this is the case with many common link-layers.

A6.1.2.2 Address Resolution

Address Resolution refers to the determination of the link-layer address of a neighbour given only its IP address. It is performed only on nodes, which are determined to be on-link by the routing table and for which the sender does not know the corresponding link-layer address.

Each node maintains a neighbour cache for mapping between IP and link-layer addresses. A node can have a unicast packet to send to a neighbour, but the cache does not hold an entry for the corresponding

link-layer address, so it has to discover the link-layer address through some mechanism. In IPv6 this is known as the Neighbor Discovery procedure as illustrated by Figure A6-3.

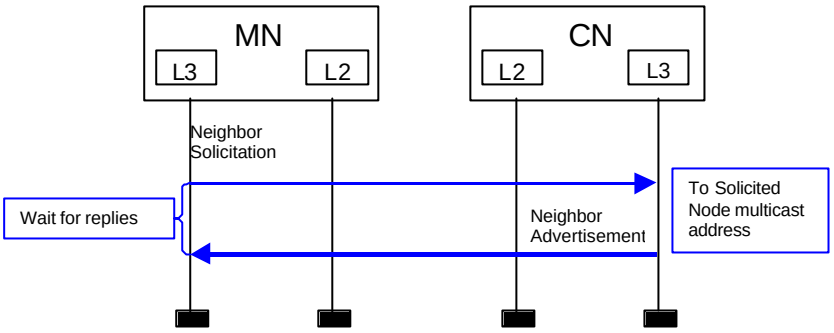


Figure A6-3: Neighbor Discovery Procedure

Node sends a Neighbor Solicitation message to the solicited-node multicast address corresponding to the target address. The neighbour (or Correspondent Node) will reply with a Neighbor Advertisement message including its link-layer address as a Source Link-Layer Address option. The Neighbor Cache is updated with this link-layer address. In future, the cached address is used instead of performing address resolution.

A6.1.2.3 Address Acquisition

An IPv6 compatible node needs an IP address of global scope if it wants to communicate with the rest of the Internet. This procedure, hereafter referred to as address acquisition, is performed when one of the interfaces attaches to a new link. Interface attached to a link when it is first brought up or while changing location from one access point to another. Also, the previous IP address might have become deprecated due to limited lifetime.

In this context it is not needed to know why address acquisition is needed. It is usually performed as described in Stateless Address Autoconfiguration [A6.4] or DHCPv6 [A6.12]. What follows is a somewhat simplified description of the operation. Address acquisition includes creating a link-local address, verifying its uniqueness on the link, and determining what information should be autoconfigured (addresses, other information, both). Addresses can either be acquired using the stateless or stateful (e.g. DHCPv6) mechanisms. This is indicated as a flag in the Router Advertisement (RA) messages. These mechanisms complete each other to provide more possibilities for the operator. Stateless approach is usually used when operator is not particularly concerned about the exact addresses, which are configured. Stateful mechanism allows for a tighter control over the nodes attached to the link.

Each IPv6 address has an associated exact lifetime (but possibly infinite) to indicate how long addresses are bound to the interface. On expiration the binding becomes invalid and the address can be assigned to another interface. Interfaces with deprecated IP addresses must use the same procedure to acquire a new IP address.

In summary the design goals of the IPv6 address acquisition facilities are; 1) Manual configuration of individual machines should not be needed. Consequently, a mechanism for obtaining unique address for each of the interfaces is needed. This mechanism assumes that interface can provide a unique (at least within the link) interface identifier. 2) Presence of a stateful server or even a router should not be needed. Plug-and-play communication with other nodes on the link is achieved by using link-local addresses. 3) Even on larger sites a stateful server should not be needed for autoconfiguring site-local or global addressed. 4) Configuration should facilitate graceful renumbering of nodes for example when changing network providers. This is achieved by leasing of addresses to interfaces and assigning of multiple addresses to the same interface. 5) Administrators need the ability to specify which autoconfiguration system, stateless or stateful, is used.

A6.1.2.3.1 Overview

Mobile nodes can either wait for the periodic unsolicited Router Advertisements or send a Router Solicitation message. Figure A6-4 illustrates the network layer signalling in the latter case.

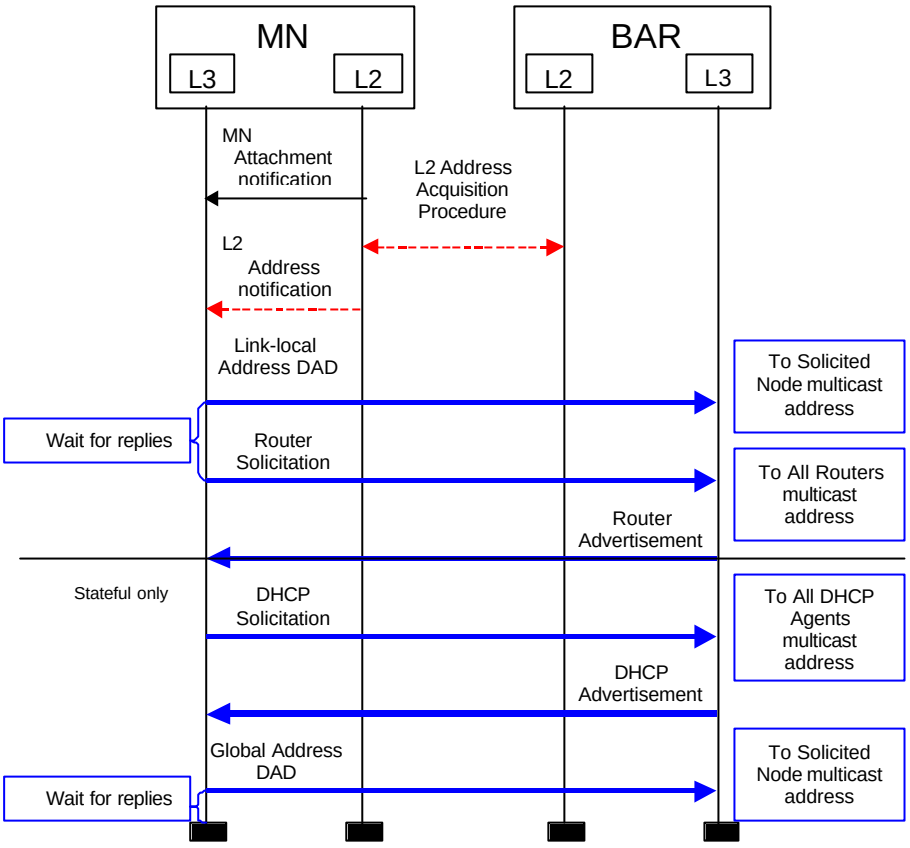


Figure A6-4: Address Acquisition Signalling

The Router Solicitation message is sent to the all-routers multicast address with the link-local address as the source address. Link-local address is created by prepending the interface identifier with the well-known link-local prefix FE80::0 [A6.3], but before using this "tentative" address, mobile node has to check that it is not already in use by using the Duplicate Address Detection (DAD) procedure. In principle the DAD procedure would have to be performed for each acquired address. However in case of Stateless Address Autoconfiguration, it is sufficient to perform this check only once for the uniqueness of the interface identifier. The same identifier is used to generate the other addresses, so their uniqueness is assured.

DAD is performed by sending "DupAddrDetectTransmits" amount of Neighbor Solicitation messages with the address being checked as the target. Source address is set to the well-known unspecified address of 0:0:0:0:0:0:0:0. The default value for "DupAddrDetectTransmits" is one (1) but can be overridden if not appropriate for the particular link. Address is considered unique if no Neighbor Advertisement messages are received from the checked address for "RetransTimer" milliseconds, default of which is 1,000 milliseconds. Regardless of specific values for these defaults, the DAD procedure clearly takes too much time. Mobile IPv6 specification relaxes the DAD procedure by allowing it to be asynchronous with respect to rest of the Autoconfiguration.

It is suggested that to prevent packet storm situations nodes should delay the transmission of Router Solicitations between zero to MAX_RTR_SOLICITATION_DELAY seconds, the default of which is one (1) seconds. Mobile IPv6 relaxes the requirement by allowing the message to be sent immediately. On links where the hardware link address does not equal the link-layer address, it is important to include it in the message as a Source Link-Layer Address option. If not included, router might not have knowledge of the link-layer address85 and it would have to use Neighbor Discovery to discover it.

Router or routers, as there can be several, respond with a Router Advertisement message. The primary contents of this message are shown in Table A6-1.

[85] With some link layers the link-layer address and the hardware interface address (if any) might be different.

Field	Description
Source Address	Must be the link-local IP address assigned to the router.
Destination Address	Source Address of an invoking Router Solicitation or the all-nodes multicast address.
Authentication Header	If a Security Association for the IP Authentication Header exists between the sender and the destination address, then the sender should include this header.
M	1-bit "Managed address configuration" flag. When set, hosts use the administered (stateful) address autoconfiguration in addition to any addresses autoconfigured using stateless address autoconfiguration.
O	1-bit "Other stateful configuration" flag. When set, hosts use the administered (stateful) protocol for autoconfiguration of other (non-address) information.
Router Lifetime	16-bit unsigned integer presenting lifetime associated with the default router in units of seconds. Lifetime of 0 indicates that the router is not a default router and should not appear on the default router list.
Source link-layer address	The link-layer address of the interface from which the Router Advertisement is sent.
MTU	Should be sent on links that have a variable MTU.
Prefix Information	These options specify the prefixes that are on-link and/or are used for address autoconfiguration. A router should include all its on-link prefixes (except the link-local prefix) so that multihomed hosts have complete prefix information about on-link destinations for the links to which they attach.

Table A6-1: Router Advertisement Message

Mobile node uses the Prefix Information options contained in the Router Advertisement messages to create site-local and global addresses. Alternatively if the "managed" flag is set, node has to use the stateful address configuration protocol to acquire the address.

Valid Router Advertisements contain one or more Prefix Information options. The most important fields of the option are explained in Table A6-2.

Field	Description
Prefix Length	The number of leading bits in the Prefix that are valid. Ranges from 0 to 128.
L	1-bit on-link flag. Indicates that this prefix can be used for on-link determination. When not set the prefix might be used for address configuration with some of the addresses belonging to the prefix being on-link and others being off-link.
A	1-bit autonomous address-configuration flag. When set indicates that this prefix can be used for autonomous address configuration.
Preferred Lifetime	The length of time in seconds that addresses generated from the prefix via stateless address autoconfiguration remain preferred.
Prefix	An IP address or a prefix of an IP address.

Table A6-2: Prefix Information Option

A6.1.2.3.2 Interface Identifier

If stateless address acquisition method is used, addresses are created by prepending the interface identifier with the prefix(es). This identifier is assumed to be a maximum of 118-bit in length (e.g. EUI-64) and unique at least within the link. In case the prefix and identifier do not add up to 128 bits, that prefix should be ignored. It must be noted that in practice prefixes and identifiers should be 64 bits in length.

Identifier can contain specific flags to give further information about its properties. Table A6-3 for example, shows the format of the EUI-64 based interface identifiers.

0-15	16-31	32-47	48-63
CCCCCCUGCCCCCCCC	CCCCCCCCMMMMMMMM	MMMMMMMMMMMMMMMM	MMMMMMMMMMMMMMMM

Table A6-3: Interface Identifier Based on EUI-64

In the table above, 'C' bits refer to the assigned company_id. 'U' bit is the universal/local flag. 'G' is the individual/group bit and 'M' bits refer to the manufacturer-selected extension identifier. If 'U' flag is set then the identifier can be assumed to be universal (or global) in scope. If the value of 'U' bit is enforced strictly, network layer could make assumptions about the necessity of DAD for example.

A6.1.2.4 Implementation Discussion

Addresses formed using IPv6 Stateless Address Autoconfiguration are a combination of the network prefixes and interface identifiers. Interface identifiers are often formed by using link-layer addresses. With some link layers these link-layer addresses are derived from hardware serial numbers or other non-changing information. This static identifier can possibly used to track the movement of the MN and the user, even if the upper layer payloads were encrypted.

Given the proposed interface, it is possible to somewhat optimise the address acquisition. These optimisations can be classified according to the network signalling they propose to eliminate:

A6.1.2.4.1 Duplicate Address Detection

The mobile node has to normally perform Duplicate Address Detection (DAD) on its link-local and other addresses. There is a possibility to avoid DAD in some cases if there are co-operating MNs and BARs. BAR can keep a database of the allocated Interface Identifiers.

The BAR and the co-operating MNs negotiate the interface identifiers in the following way. MN forms a new IID in an unspecified manner. It uses the IP2W_PROPOSE_IID request to propose the IID to the BAR. The IID is sent in a link-specific manner to the AP (or similar entity). IP2W_IID_PROPOSAL indication used to inform the BAR about the IID proposal. If the address is not yet allocated, the proposal can succeed and the response can contain a positive acknowledgement. The allocated IIDs are kept in a database keyed by the link-layer address of the mobile node. These entries can be removed if the corresponding mobile node detaches. BAR should avoid Denial of Service type situations by allowing only one or two interface identifier per single link-layer address (or mobile node).

Those mobile nodes that do not support this optimised procedure must perform DAD as specified. BAR can then possibly defend the link-local addresses on behalf of the mobile nodes by replying with Neighbor Advertisement messages.

A6.1.2.4.2 Router Solicitation

With certain link-layers, there is a central authority (or access point) which has complete knowledge of the nodes on the link. In these kinds of situations, the IP2W_NODE_ATTACH indication can be used to indicate higher layers about new nodes on the link. Implementation can either place the access point in the BRAIN Access Router or access point can act as a proxy for the BAR. Higher layers react to this information by sending the Router Advertisement message to the link-layer address of the newly attached node.

IP_2W Configuration Interface should present this capability as a flag. In case the flag is set, mobile nodes should act according to the Stateless Address Autoconfiguration [A6.4] and wait for up to MAX_RTR_SOLICITATION_DELAY before sending the Router Solicitation. If not set, then Mobile IPv6 behaviour should be used instead. The wait is carried out to prevent excess signalling, which why the default delay should be set based on the capabilities of the link to a possibly quite small value. The Router Solicitation can be cancelled if Mobile Node receives the RA before the wait is over.

It must be noted that this optimisation should be implemented only if there is a measurable advantage to it.

A6.1.3 Quality-of-Service Control

The provision of QoS within the BRAIN Access Network is a major issue since the BRAIN technology is designed to be used for instance by voice and multimedia applications (which means real time, or very sensitive traffic). QoS is mostly studied at both the session/application layers (with protocols like SIP, RTSP, RTP) and at the network/transport layers (with architectures like DiffServ or IntServ with its associated signalling protocol RSVP).

Whatever choices will be made for BRAIN, the QoS cannot be a strictly high layers issue. Indeed the link layer must not undo the QoS related scheduling and processing the network and upper layers have performed. This means that the link layer will have to perform some QoS processes related with the ones performed at IP layer in order to maintain on the radio link the QoS requirements coming from the user (in the high layers of the equipment for instance).

As a result, there will be an interaction between the IP QoS and the Link Layer QoS, which the IP_2W interface will have to deal with.

Concerning the BRAIN system, many QoS scenarios have to be considered because the IP layer can manage QoS in many ways. Moreover the QoS management at the link layer is quite free as well because of the possible enhancement of the convergence layer based on the HIPERLAN/2 standard, for instance.

A6.1.3.1 Basic Requirements for the IP_2W QoS

When a QoS context is created at the IP layer, it will have to be taken into account at layer 2, in order to carry some QoS-constrained data on the radio link. We consider in the following that the convergence layer is QoS capable, that it is able to map IP QoS contexts to the link layer QoS contexts.

The IP_2W interface is separated into a Data Interface and a Control Interface. The QoS issue is quite different in these two planes; it mostly concerns the Control Interface.

A6.1.3.1.1 Control Interface: QoS Control

The QoS Control refers to the creation of a QoS context at the link layer, and to all QoS information necessary to maintain a defined QoS apart from the data themselves: information about the QoS capabilities of the link layer or about QoS changes (improvement or violation).

?? **Establishment of the L2 QoS context**: before transmitting QoS-constrained packets, the IP layer has to check if the link layer will be able to respect the QoS. If so the link layer has to grant a QoS context to the packets. That means the need for messages crossing the interface for QoS requests from L3 to L2, to convey mappings between L3 and L2 QoS, and the answer from L2 to L3**.**

?? **Advertising of the QoS capabilities**: on request from L3, the link layer should be able to advertise its QoS capabilities. We need here another exchange of messages between both layers.

?? **Changes in QoS**: during the transmission of data from the same application there might be a QoS change on the link layer (QoS degradation as well as QoS improvement). Again we need a message from L2 informing L3 of such a change, or L3 can query L2 for more available resources.

A6.1.3.1.2 Data Interface: QoS Support

The QoS Support concerns the data themselves, that is the transmission of IP packets, which require particular QoS parameters, once the mapping between the L2 QoS context and the IP QoS has been done. At this stage, the QoS is an IP_2W Data Interface issue, but does not highly affect the definition of the IP_2W interface. Indeed, the relevant parameters concerning the L2 QoS context are known by the IP layer and simply added as the QoS context identifier to the information attached to an IP packet transmission primitive.

A6.1.3.2 Existing IP QoS Management Proposals

As far as IP QoS is concerned, there are two primary QoS architectures to provide differentiated service to flows. Additionally some adaptation can be provided in higher layers, with the Real-Time Transport Protocol [A6.13].

A6.1.3.2.1 DiffServ

The Differentiated Services (DiffServ) architecture [A6.5], defines a model whereby network service providers can provide different 'classes' of service to traffic flows, based on bilateral Service Level Agreements (SLAs) between the customer and the provider. The basis of the model is the following:

Core routers in an IP network forward traffic in a simple fashion based on a per-hop-behaviour (PHB) associated with each IP packet, and that the more complex forwarding decisions (classification, routing, queuing, marking, etc.) are made at the network edge routers where traffic is assumed to be lighter and the number of flows smaller. Packets are routed in an aggregated fashion based on a DiffServ Code Point (DSCP). This is a value that is encoded in the DS field which supersedes 6 bits of the IPv4 TOS and the IPv6 Traffic Class fields in the IP packet header. The IP packets entering the DS capable network are marked by a boundary router and inside the network, in each DS-capable router, a table is used in order to map the DSCP with the PHB which need to be received by the packet, if this PHB has been implemented in the node. Else there is default PHB in each node, which consists in the best-effort forwarding behaviour.

A6.1.3.2.2 IntServ/RSVP

The Integrated Services, [A6.6]/ Resource ReSerVation Protocol, [A6.7] (IntServ/RSVP) model [A6.8] provides enhanced QoS to applications by reserving network resources on a per-flow basis. Before data

can be received with enhanced QoS, the applications must first set up an end-to-end path and reserve resources by means of the signalling protocol RSVP. RSVP provides three basic levels of service: apart from the best-effort service, IntServ specifies Guaranteed Service (for applications with stringent real-time delivery requirements) and Controlled-Load Service (which offers the flow a service equivalent to that seen by a best-effort flow on a lightly loaded network). "Best effort" traffic shares the non-reserved network resources, leaving QoS traffic-flows undisturbed.

RSVP sends a Path message from the sender that contains the Traffic Specification (Tspec field) information to the receiver. The receiver then sends a reservation request (RESV) back to the sender, using the same route as the path message. The RESV message includes the Tspec field again and a request specification (Rspec) indicating the type of IntServ required (Guaranteed or Controlled Load) Thanks to these parameters, the data flow is characterised at the IP layer in each RSVP-enable router on the way back to the sender.

A6.1.3.3 L2 QoS

Concerning the Link Layer QoS, most of the work has been done around the IEEE 802 LAN technology. The IEEE 802.1p, 802.1Q and 802.1D standards define how Ethernet switches can classify frames in order to expedite delivery of time-critical traffic. The IEEE 802.1p standard provides priority mechanisms to enable QoS in 802- style LAN. It allows bridges on the LAN to mark a packet for QoS. The standard defines eight different priority levels and describes which type of traffic is expected to be carried in this priority, as shown in Table A6-4.

User priority	Traffic type	Comments
1	Background (BK)	
2		Reserved for future use
0 (Default)	Best Effort (BE)	Default LAN traffic
3	Excellent Effort (EE)	For valued customers
4	Controlled Load (CL)	Traffic will have to conform to some form of Higher Layer admission control
5	Video (VI)	< 100 ms delay and jitter
6	Voice (VO)	< 10 ms delay and jitter
7	Network Control (NC)	

Table A6-4: IEEE 802.1p Traffic Types

802.1D bridges support a number of queues and the user priorities are mapped one-to-one or many-to-one to queues, depending on the number of queues supported. When a packet arrives at the bridge, it is added to a queue according to its priority marking. The user priority information is carried in each IEEE 802 frame using a Tag Header following the source and destination address.

This mechanism allows different behaviours at layer 2 depending on the priority of the frame, so that it performs a basic QoS management.

A6.1.3.4 QoS Architecture Combining L2 and L3 Mechanisms

A6.1.3.4.1 Subnet Bandwidth Manager

Some work has been done in this area by the ISSLL Working Group of the IETF, [A6.9], in order to deal with the IntServ architecture over IEEE 802-style networks. This work has resulted in the development of the Subnet Bandwidth Manager (SBM) for shared or switched 802 LANs. SBM is a signalling protocol for RSVP-based admission control over IEEE 802-style networks. It provides a method for mapping an internet-level set-up protocol such as RSVP onto IEEE 802-style networks. In particular, it describes the operation of RSVP-enabled hosts/routers and link layer devices (switches, bridges) to support reservation of LAN resources for RSVP-enabled data flows.

Basically, the SBM protocol performs at layer 2 the same functions as RSVP does at layer 3. In order to perform this, two primary components are required:

?? a Bandwidth Allocator (BA) maintains state about allocation of resources on the subnet and performs admission control

?? a Requestor Module (RM), in every end-station, performs the mapping between higher layer QoS protocol parameters and layer 2 priority levels

Two different SBM architectures are proposed, depending on the number of BAs per segment (Centralised architecture if there is only one BA, which must have some knowledge of layer 2 topology of the subnet, distributed otherwise.), as shown on Figure A6-5 and Figure A6-6.

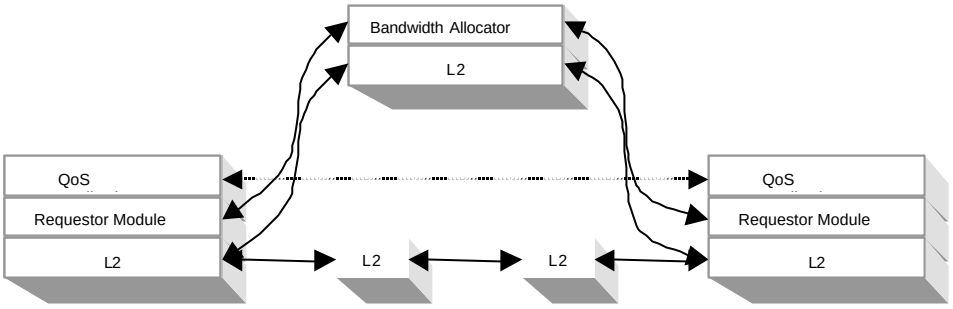


Figure A6-5: Centralised BA Architecture

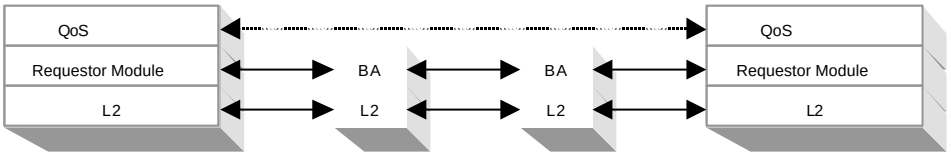


Figure A6-6: Distributed BA Architecture

In any case, the SBM protocol implies two types of communications between the different components:

?? communication between the higher layers and the RM (for the application to initiate, change or delete reservations, for the RM to inform the higher layers of a QoS unavailability, etc.)

?? communication between the RM and the BA, or between BAs (a signalling mechanism similar to RSVP)

The details of this protocol are not fully developed here since the aim of this paper is to give an overview of the work done in the area in order to find out what could be done in the BRAIN context.

A6.1.3.5 IP_2W QoS Proposals

It is supposed first that the Link Layer can support a number of QoS context otherwise there is no need trying to perform any QoS at this level, even if some QoS protocols exist at the upper layers. (This requirement is achieved in the HIPERLAN/2 standard, since the Ethernet SSCS can support the IEEE 802.1p based priority scheme.) The definition and implementation of those contexts in the nodes of the network are part of the work to do at the convergence layer. We focus here on the consequences for the IP_2W interface.

A6.1.3.5.1 QoS Control (Control Interface)

A6.1.3.5.1.1 IntServ Context at the IP layer

?? QoS request

At the Control Interface, the IP layer has packets from the same flow to send with a number of QoS parameters. L3 has thus first to ask for a QoS context ID on L2, that would fit with those parameters. This request will basically consist in a primitive, e.g. QoS Request, meaning "I would like a QoS context for a traffic that conforms the following <FlowSpec>". The FlowSpec would thus be one of the parameters for this primitive and is similar to the Guaranteed Service IntServ specification.

?? QoS request answer

At this stage, the L2 has to find a QoS context able to perform the QoS needed by the upper layers (function provided at the CL), and keep inform L3. A QoS Context ID of the mapping between the IntServ QoS definition and a L2 QoS context will be passed to L3 as the answer to the previous primitive.

If the link layer is not able to respect the <FlowSpec>, for example, there is not enough bandwidth, it will return an error.

Once L3 knows the QoS context that will be used for its flow at L2, it will start transmitting the data packets of the flow using the Context ID in each packet. Enforcement of the QoS is here managed by the Data Interface (see Section A6.2.9).

If explicit reservations are not available at the link layer, the IP layer with know about this, since it can get the properties of the link layer with a capabilities query. Thus, it will use some priority-based QoS Context for the IntServ flow and schedule the packets itself.

A6.1.3.5.1.2 DiffServ context at the IP layer

With DiffServ, several flows may be put within the same forwarding treatment group. Packets within a service class are independent from each other. The requested behaviour of packets in a flow is indicated with the QoS Context ID that is passed with each packet; this is a Data Interface function.

A6.1.3.5.1.3 Change of QoS

?? QoS violation

The link layer is supposed to support the context transmitted by the IP layer. But the resources available at a node can change so that a constraining context cannot be supported anymore. The link layer is the one which can detect a more permanent (compared to a short but sudden) QoS violation because it is aware of both QoS contexts needed and actually granted to this application and will thus have to inform the IP layer of any violation. QoS violations are applicable only to reservation based link layer flows, thus other QoS violations need to be noticed at the IP layer.

A further complication can happen after a handover. Let's assume that the mobile had five 100 kbps reservations for flows. If the new AP can only support 300 kbps of reserved bandwidth, some entity must make a decision about whether all five flows will be downgraded to 60 kbps or two flows will be closed in favour of the other three.

QoS violations are indicated to the IP layer through a single primitive that covers all the ongoing flows. As a result, all reservation-based flows are deprecated. The IP layer must then make the decision about which flows it will try to get the resources back. In the example with five flows, the IP layer can start to request resources, first for the most important flow (according to some user/application driven priority) and then continue until the link layer cannot allocate resources to further flows.

The Outage flag in the data interface can also be used to inform of momentary resource problems. When problems arise on the link and packets are difficult or impossible to get delivered, the link layer can use the Outage flag in the confirmation-messages of the send-primitive: when link problems are ongoing, the bit is set, otherwise it is zero. A link outage is different from more general resource outage in the whole cell. A link outage is seen from the single MN that receives the indication.

An open question still remains. What must the BAR do, when the link layer informs of a resource outage for the downlink; the mobile node and its user are the most capable to decide on which flows will be re-reserved resources.

?? QoS improvement

In certain situations, a mobile might want to increase the reserved resources, for example to get a better quality video stream, which was not possible when the flow was initially set up. Thus, the link layer should be able to indicate that new resources are now available at the node. This would require asynchronous messages from L2 to L3. Another option would be to add a primitive, which the IP layer can use periodically to query for more available bandwidth.

In the IP_2W interface, the IP layer can poll for new capacity by using the primitive to request a reserved bandwidth. It can set an existing QoS context identifier in the call and set a larger bandwidth, for example. If resources are available, the link layer will make the modification and return a positive signal. However, if resources are not available, the return value indicates this. This is only applicable if the link layer can provide explicit reservations.

However, requesting an upgrade in the resources is less straightforward for the downlink direction. It is an open question how a MN can trigger the BAR to initiate to request, for example, more bandwidth for an incoming flow.

A6.1.3.5.1.4 Advertising QoS capabilities

On request from L3 the link layer should be able to advertise its QoS capabilities. Actually the request is originating from L3, this feature would thus be performed using a primitive from L3 to L2, e.g. Query QoS Capabilities, meaning "Which QoS characteristics can you support on the link layer?". This primitive can have many optional parameters, corresponding to which parameters L3 is interested in. According to the return values, the IP layer will know about the properties of the link layer and can therefore use the proper primitives and scheduling mechanisms to handle the different QoS-sensitive flows.

A6.1.3.5.2 QoS Support (Data Interface)

A6.1.3.5.2.1 IntServ Context at the IP layer

Once the QoS control functions have been achieved the IntServ flow will be delivered to the link layer (that is the data packets). Following the QoS Request primitive mentioned in section A6.2.4.1, the IP layer has been returned a QoS context identifier for its flow. This QoS context will be passed with each packet of the flow to give information to the CL about which link layer queue each packet should be put to. L3 must be able to differentiate the flows and set the proper identifier to each packet.

A6.1.3.5.2.2 DiffServ context at the IP layer

This case is quite similar to the previous one except the QoS identifier will differ. Similarly, the IP layer first requests a QoS context identifier from the CL, which is then used in each data packet provided to the CL to indicate the proper link layer queue.

A6.1.3.5.2.3 No QoS guarantees

If the IP layer does not specify any QoS guarantee, the QoS context identifier is empty. There is still a mapping to do at the CL between such packets and the identifier and queue corresponding to the Best Effort context (for instance the priority 0 if the CL uses the 802.1p priorities scheme).

A6.1.3.5.3 Scheduling Interaction

Traditionally, operating systems have implemented a quite simple data interface between the network and link layers. Conceptually, each network device has a single input queue. The device is assumed to transfer packets in first-in/first-out order, and not hold anymore packets than is necessary for the hardware to function. The layers perform flow control by sending XON/XOFF like flow-control indications to each other.

As this model is so simple, it can be applied in practically all scenarios. However, to get better performance and robustness, a more sophisticated model might be justified. Specifically, the wireless environment seems to demand it. The wireless medium cannot be adequately abstracted as a single channel:

?? There can be multiple base stations or transceivers behind a single network device. Effectively, there are multiple independent schedulable channels that have to be serviced through a single point of congestion.

?? The wireless channel suffers from a relative high error rate. In itself, that is not harmful. However, these errors are location-dependent and possibly bursty. Mobile nodes experience error rates that are largely independent of other traffic on the link. So flows that are destined to a particular mobile node may or may not be able to make progress, regardless of other flows destined to other mobile nodes. Congestion is therefore mobile node dependent.

?? Many wireless link layers can perform forward and backward error correction. These are techniques that can somewhat alleviate the effects of the wireless environment. At the same time, they complicate the scheduling and increase the buffering on the link layer.

?? Packet flows have different reliability and delay requirements, which should be reflected in the link layer and re-transmission scheme of the radio hardware.

Consequently, applying the traditional interface in a wireless environment poses several problems. Link layer needs a possibly large set of schedulable packets, to select which to send next. This significantly reduces the possibilities of the network layer to control traffic flows.

Thus, the traditional scheduling model is inadequate, alternative models should be considered. It can be argued that the best way to proceed is to determine the respective roles of both layers. As the Figure A6-7 illustrates, there are three primary choices for the scheduling model. The decoupled approach (a) resembles most the model that was described above. The network-controlled (b) and link-controlled (c) approaches are described below.

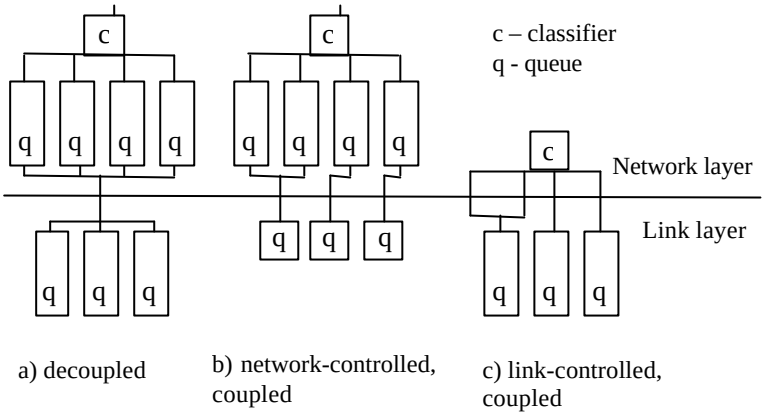


Figure A6-7: Primary Scheduling Model Choices

In the link-controlled approach the link layer handles most of the scheduling activities. The arguments against this model are significant:

?? To avoid considerable layer violations, the data interface still needs to be changed. Link layer can not perform packet classification.
?? Support for many advanced scheduling functions, such as Explicit Congestion Notification (ECN) [A6.10], cannot be achieved without layer violations.
?? Network layer scheduler needs to be implemented anyway, for other link layers. Having each link layer implement this functionality results in needless code duplication.

The last choice left is the network-controlled, coupled approach. Requirements for the interface are:

?? Link layers should be able to make optimal use of all their features. This includes capabilities such as link state prediction and monitoring, error detection, and forward and backward error correction. For example, the link layer should be able to decide a set of eligible flows, and decide the next packet to send from this set. The set of eligible flows can possibly be the set of flows that can make progress.

?? Link layer characteristics such as centralised scheduling, or collision handling in the distributed approach, should be handled optimally. In the former case, the link layer needs to know beforehand the set of packets that can be transmitted in the near future, in order to make educated guess for the resource request.

?? Granularity of flows, as evidenced by the link layer, should be low enough that the burden for the link layer is minimal. However, granularity should be high enough, that no flow should contain both packets that can make progress and packets that can not make progress.

?? Network layer scheduler should have all the information and control capabilities needed to provide guaranteed performance. This includes having absolute control over how long a packet should be queued, and what should happen to those packets that exceed this bound. The network layer should be made aware of link layer conditions, such as whether flows can make progress or not.

Based on these requirements, we formulate the respective roles of the layers in the IP_2W scheduling model.

A6.1.3.5.3.1 Network-layer Scheduler

The network layer scheduler is the controlling entity in the scheduling model. In principle, its task is to ensure that all flows receive the desired service. This service is characterised by a service curve. Parekh

and Gallager [A6.11] introduce the concept to abstract the behaviour of particular scheduling algorithms. The service curve is such, that given an arrival function $A_i(t)$ for flow i, the output function $S_i(t)$ can satisfy certain bounds for the flow.

Network layer scheduler controls the buffering of backlogged packets. Link layer may inquire the status of the individual flows, to gather context for its own decisions. However, link layer should allow only for minimum amount of packets to be buffered on the link layer. The minimum is determined to be the amount of packets, that is enough for the hardware to function optimally.

A6.1.3.5.3.2 Link-layer Scheduler

The task of the link layer is to ensure that the wireless link functions optimally. It should not contradict with the fairness of the service, as provided by the network layer scheduler. Link layer should not implement policy by itself, but be guided by the control interface to make decisions regarding error correction and other advanced features.

The link layer co-operates with the network layer scheduler to provide service that aspires to hide the effects of the wireless link. Network layer can allow limited leeway for the link layer, on packet-per-packet basis, to attempt through retransmissions and other means, to deliver the packet. Control interface should have means to specify how rigorously the delivery of the packet should be pursued. Effectively, the interface should unambiguously express, which packet should be preferred in the case, where two contest for the same resource. That includes the case whether to retransmit now or later, if there are other deliverable packets.

In some cases these guidelines cannot be followed. Namely, the proposed scheduling model assumes that the wireless link, i.e. the congestion point, is directly connected to the entity containing the network layer scheduler. In cases, where there is a large link layer infrastructure, e.g. base station subsystem, behind the ”last-hop” network layer entity, the point of congestion can possibly be quite far away. For these systems, the link layer has no choice but to perform scheduling in the link layer entities between the wireless link and the network layer scheduler.

A6.1.3.5.3.3 Notes on IP_2W QoS in BRAIN

From the different scheduling models in Figure 7 only the network- and link-controlled coupled models are feasible if the IP_2W interface is used; the decoupled model is not possible, since the IP_2W layer is aware of the link layer mechanisms can do the proper mapping from L3 to L2.

The HIPERLAN/2 link layer provides both reservation of bandwidth and relative priority without explicit resource reservations. These two service types map very straightforward to the BRAIN QoS architecture, where both IntServ and RSVP based reservations and DiffServ relative priorities provide a wide range of services. Since BRAIN is based on IP technology, we would suggest using the network-controlled coupled approach for scheduling IP packets. An important optimisation task would be to define the amount of data buffered at L2. The link layer must be able to function at full capacity, but not store too much data, so that the QoS noticed at the IP layer is not compromised.

A6.1.4 Handover Control

IP mobility protocols have traditionally tried to avoid making assumptions on the link layer between the MN and an access network when specifying the mechanisms for access router discovery and movement detection. However, it has been noticed that the anticipation of prospective handover to a new router is essential in achieving fast and smooth handovers. In the recent handover proposals at the IETF, this need for anticipation is expressed as an assumption of the availability of link layer “triggers” that can accelerate IP level handover procedures. Handover control in IP_2W tries to capture these needs for signalling of handover events to upper layers and for initiating handovers to new access routers.

A handover control interface should be applicable to a variety of handover scenarios. Depending on the wireless technology, local policies, or points of view of different protocol layers, handovers can be classified into several types:

?? When handovers can be anticipated, the old and new access routers can negotiate on establishing service for the MN at the new router. These handovers are planned or proactive (as opposed to unplanned and reactive handovers). This is a network layer view where “planning” typically expects the link layer to give hints of imminent handovers.

?? Handovers may be initiated by the MN (Mobile-controlled handover, MCHO) or by the access network (Network-controlled handover, NCHO). Furthermore, the MN or the network may assist each other in making the handover decision (Mobile-assisted handover, MAHO or Network-assisted handover. NAHO).

?? Some technologies allow the MN to be simultaneously connected to the old and new access routers. Handovers are then soft (make-before-break). These are typically backward handovers. Otherwise the handover is hard (break-before-make), and typically forward.

?? A radio handover occurs entirely within the link layer without the MN changing the BAR. A network handover within the BAN occurs between BARs. IP_2W functionality is mainly related to network handovers. However, even a radio handover may have implications on QoS provision.

In this context, handover control not only includes getting hints on imminent handover events (e.g., for facilitating movement detection in the MN or proactive operation in the network) but also the support for making decisions on performing handovers. Generic high level support for handover control procedures is looked into. It would not be dependent on specific wireless link layer technologies. For more detailed control, an advanced but also more technology specific control interface would need to be specified. In particular, radio signal measurement and resource availability signalling for network resource management is not considered here, although IP_2W might provide a generic message passing function for conveying requests for and reports on radio resources and measurements. We assume that the link layer is able to independently monitor the radio link quality and start measurements on neighbouring radio transmitters when the link quality degrades between a BAR and its MNs. If a BAR selection mechanism is to be specified which relies on link layer measurements, current load on the BARs, etc., the protocols for conveying and algorithms for processing the needed information have to be specified. Harmonising these protocols for diverse radio technologies is essential for achieving a generalised radio and BAN resource control. However, these mechanisms for network resource control are beyond the scope of the IP_2W itself.

At a handover, part of the MN's link-layer context may need to be transferred from the old access router to the new access router using network-layer signalling. Therefore, IP_2W must provide means for pulling the MN's context up at the old access router and pushing it down at the new access router. The context information may include header compression and QoS states, and encryption keys, for example.

This section proposes the IP_2W functionality (neighbourhood awareness, handover progress monitoring, and handover decision control) which forms the basic building blocks for handover mechanisms. Handover control adds to and builds on association control (i.e., connection control) which includes the basic node attachment/detachment (connection/disconnection) primitives and which is a mandatory subpart of the overall handover control.

Although handover control is asymmetric in nature, here we specify handover management collectively from the MN and from the BAR point of view. The requirements for handover control in the MN and the BAR differ, but the eventual handover support mechanisms may be very similar in both cases. The exact parameters used to trigger a handover and the actual decision process, however, are beyond the scope of IP_2W.

A6.1.4.1 Neighbourhood Awareness

When enabled, neighbourhood awareness provides a view of nearby BARs that are candidates for handover, ranked in the order of preference. Each entry is accompanied with a value indicating the "goodness" of a particular BAR, based on an arbitrary scale that is independent of the wireless link technology.

The ordered list of nearby transmitters contains an identification of candidate BARs. The BARs are identified by their hardware addresses, IP addresses, or NAIs. The MN may use this information for its own handover decisions in MCHOs. This list is passed to the upper layers through an event notification. In MAHOs, the list may be conveyed from the MN to a BAR to aid the BAN in making handover decisions. A BAR may also receive this information from neighbouring BARs.

The information provided by neighbourhood awareness depends on the internal procedures of the link layer and may be unavailable except at well defined times. It is assumed that the information is complete and available at least when the link layer is ready to transition from handover preparation phase to handover execution phase.

A6.1.4.2 Handover Progress Monitoring

When enabled, handover progress monitoring allows monitoring of current handover phase (in a BAR, for each attaching or detaching MN), BAR selection, and handover timing. In the MN, handover progress notifications have a direct relationship with move detection. When using NCHO, the MN only receives notifications on handovers but it cannot decide on these handovers. These notifications are accompanied with the identity of the new BAR, such as with its hardware address, IP address or NAI (Network Access Identifier). The IP_2W signals the upper layers of the handover phase via an event notification. In the BAR, handover progress events may be used for triggering fast and smooth handover mechanisms on the upper layers.

The following events separating different handover phases can be identified:

?? A handover is suggested (e.g. the signal quality has dropped below a certain threshold). This signals a proposition to move to handover preparation phase.

?? Candidate BARs have been selected for handover (the list and the preferred selection is available via the neighbourhood awareness function). This signal enables proceeding to handover execution phase.

?? MN has detached from old (or this) BAR.

?? MN is attaching to this BAR (may be rejected by BAR)

?? MN has attached to new (or this) BAR

?? Handover is completed. Signals handover completion (either success or failure). This assumes that the link layer supports the notion of handover. Otherwise, we may simply get the notification that the MN has attached to new BAR (or back to old BAR in case of failure).

?? an attachment request has been rejected.

The order of the attachment and detachment events is not fixed. A MN that supports soft handover may first attach to the new BAR and then detach from the old one.

The upper layers may take specific actions depending on the handover phase. For example, the events may initiate upstream buffering or advance registration procedures at the MN. These mechanisms vary depending on the fast handover schemes and they are beyond the scope of IP_2W.

A6.1.4.3 Handover Decision Control

When enabled, handover decision control allows control over the BAR selection and over the exact timing of the handover phases. For controlling BAR selection, neighbourhood awareness is required. For controlling handover timing, handover progress monitoring is also required. When the handover progress monitoring function signals that the previous handover phase has been completed, the handover decision control function can be invoked to allow the handover to proceed into its next phase. Handover decision control at the BAR is only available for NCHO (and MAHO), and at the MN it is only available for MCHO (and NAHO).

The possible actions for allowing the link layer to progress from one handover phase to another are:

?? Prepare for handover or tell the MN to prepare for handover (e.g. look for candidate BARs for handover). This commands the MN to enter handover preparation phase.

?? Execute handover to new BAR. This action assumes that the link layer supports the notion of handover. Alternatively, the handover can be decomposed into successive detach and attach actions.

?? Detach from old BAR

?? Attach to new BAR or tell the MN to attach to new BAR (optionally, the new BAR may be selected from a list provided by the neighbourhood awareness function)

?? Accept or reject a MN that is attempting to handover to this BAR. Note that the MN may also be rejected independently by the admission control function as part of security management or QoS control if these are not supported as an integral part of the handover.

Again, attach and detach may occur in any order, depending on the handover type supported by the link layer.

The handover control functionality presented here overlaps with radio and network resource control activities. The role of IP_2W handover decision control is to override, time or tune the resource control actions.

A6.1.4.4 Example Application

Handover protocols can be specified by using very generic network-layer mechanisms. Although a generalised handover procedure is supposed to be fully handled at the network layer, some issues concerning the link layer still remain. Therefore, network-layer solution cannot exist without any communication between the different layers.

Figure A6-8 illustrates handover signalling in a planned MCHO scenario where the MN detaches from an old access router (OAR) before connecting to the new router (NAR). The arrows in the figure represent network-layer messages. The Host Handover Request message instructs OAR to start building a tunnel to NAR for packet forwarding (possibly using bi-casting) and transferring any required context of the MN to NAR. The Router Advertisement message notifies the MN of the availability of the NAR for registering.

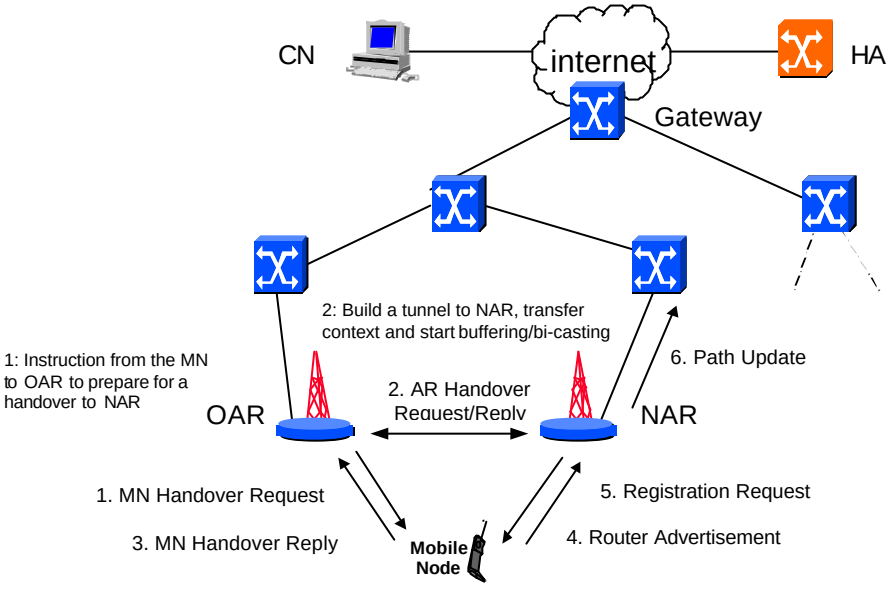


Figure A6-8: Planned MCHO Signalling

From the message sequence chart in Figure A6-9 we can notice how handover event notifications at the IP_2W interface may facilitate proactive IP handover preparation between the access routers, and how they can trigger the mobility registration procedure at the new access router. That is, the network-layer Host Handover Request and Router Advertisement messages shown in Figure A6-8 can be triggered by the handover responses and indications at the IP_2W interface.

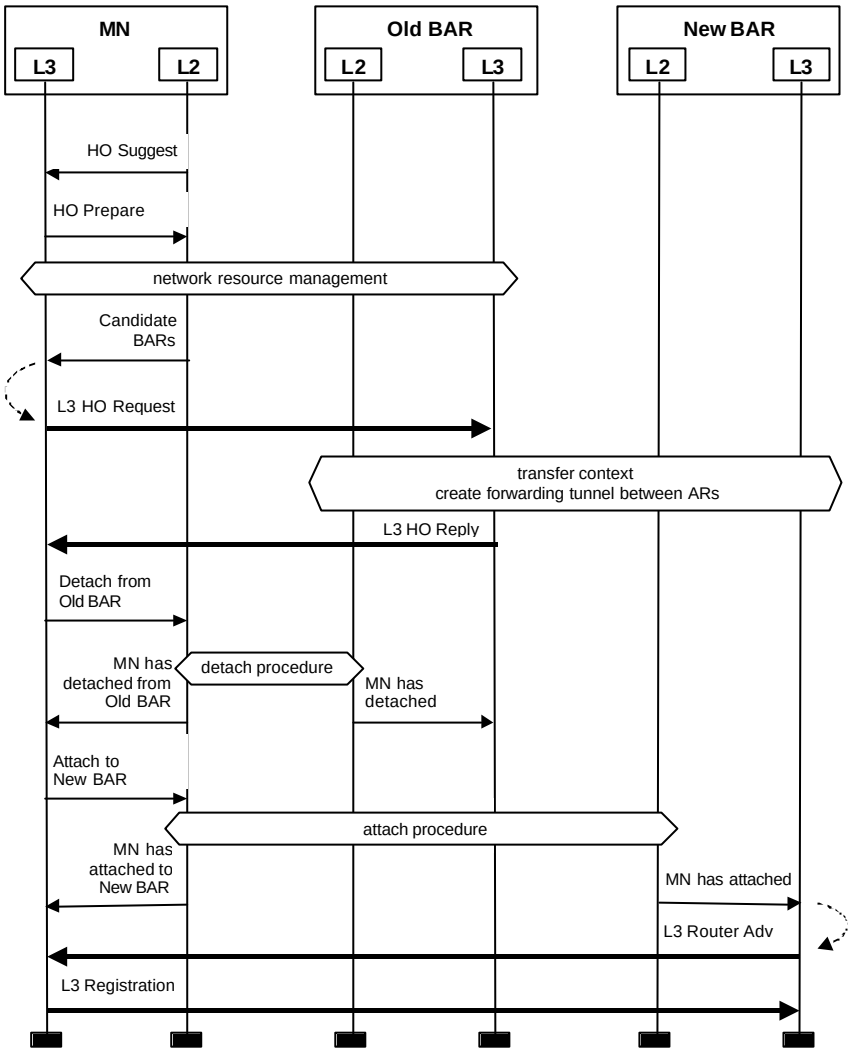


Figure A6-9: A Mobile-Controlled Planned Handover

A6.1.5 Idle Mode Support

A6.1.5.1 Idle/Standby Modes and Paging

A MN supports different operating modes in order to optimise its mobility support. While the MN is powered off, out of the coverage area or does not require any connectivity with the network, it is considered as being detached. Otherwise, the MN may be in active or in idle mode. In the active mode, the MN is sending and or receiving packets via its interface to a network. In the idle mode, the MN is not transferring IP data packets. Furthermore, if the MN is a terminal that supports the Advanced Configuration and Power Interface (ACPI, APM), the MN or subparts of it may be sleeping in a standby mode. Then, the MN may be woken up by an external signal. This signal may be the reception of an IP packet, for example. The state-changes may also be explicitly initiated by network or terminal command (e.g., by a paging request received from the network or the need for the terminal to initiate location updates).

In summary, there can be identified two separate but interrelated concepts of state with respect to the mode of activity, which can be coarsely characterised as follows:

?? active/idle/detached according to IP packet transmission activity

?? active/standby according to (link layer) power management

Hence, idle mode reduces signalling load, and saves radio spectrum and routing state in the BAN whereas standby mode saves battery in the MN. These two concepts may coincide in a MN but not necessarily; an idle MN need not be in standby mode even if power management is supported.

When a MN enters the idle mode it records the identity of its current paging area (or several of them if there are overlapping areas). A paging area consists of a set of BARs. When the MN migrates within the BAN it informs the BAN of its location only when it crosses a boundary of a paging area. In the idle mode, the terminal can move inside a paging area without signalling its exact location to the network (except to refresh paging timers if required by the paging scheme). The network uses paging when it needs to force the MN to switch from the idle mode to the active mode (to be able to send data packets to the MN): the BARs in the current paging area send a paging signal in order to locate the MN. When a MN is in idle mode it listens to paging signals from BARs and responds with a location update when it is paged.

The IP_2W Idle Mode support allows location tracking of idle MNs and power saving at standby mode. In the implementations, standby mode can be maintained during paging only if the paging protocol engine is not reliant on the CPU or other main resources of the MN. This is typically not possible if location management is always performed at the network layer. In the standby mode, the radio receiver at the MN may be periodically shut down provided that the access point and the MN can agree on time slots during which the MN must monitor paging area advertisements and paging signals. When the link layer detects that it does not receive the paging channel it has to wake up to be able to synchronise its radio transceiver with the periodic paging time slots.

Figure A6-10 shows a topology of the network with two paging areas. Assuming the mobile node is in the idle state, in Paging Area 1, routers only know the paging area the mobile is in. A router has to know the exact BAR the mobile node can hear, to route packets correctly. It buffers the data packets and then performs paging procedure by broadcasting a paging message to the Paging Area 1, and waits for a response before routing the data packets.

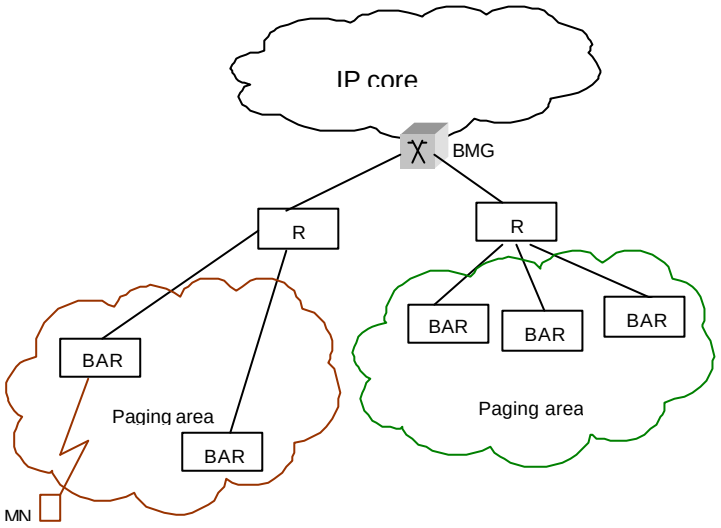


Figure A6-10: Paging Areas in the BRAIN Access Network

Performing such paging over the wireless link on network layer will unfortunately require maintaining network layer connectivity, at least the ability to receive signalling IP packets broadcast (or multicast) by the network. This does not allow the MN to enter a standby mode, as it requires computation on the IP layer.

A paging scheme should be able to use link layer signalling over the radio link. It seems to be more efficient to handle paging over the radio link by defining a broadcast or multicast paging channel. The MN gets all the distributed paging requests via that channel and determines if it is relevant for the terminal (e.g. based on an identifier of the MN included in the paging packet parameters). Then, the MN wakes up through internal mechanisms.

A6.1.5.2 Paging on the Link Layer

The short description of the paging functionality given in the previous section should be enough to point out the necessary paging messages.

Basically what is needed in a paging procedure at the link layer:

?? Identifier to each paging area.

?? Way for the mobile node in the idle state to find out which paging area it is in. One approach is to have BAR broadcast beacon messages periodically to mobile nodes. In this case, each BAR transmits its Paging Area Identifier in its periodic beacon signals, thus enabling a mobile node to notice when it moves into a new Paging Area. If a BAR belongs to several overlapping Paging Areas, it may advertise all Paging Areas in the beacons.

?? Identifier for the MN

?? Paging request initiated by a BAR. This packet must contain the identifier of the MN being searched for. Like the Paging Area Identifier, this paging request should also be conveyed in the BAR broadcast beacon signals.

To perform paging at the link layer, there is a need to make the link layer procedures fit with the IP network. This is not obvious because the link layer messages will depend on the technology used on this layer. Adaptations have to be done in order to come up with a generic paging procedure. It will consist in defining an interface between the link and the network layers, so that IP paging packets can be converted in suitable link layer messages and link layer messages can be made suitable to the IP layer (which will be the role of the convergence layer). A set of messages is needed, which will go through the interface.

The interface is asymmetric: the MN and the BAR see different primitives and messages crossing the interface. To illustrate, Figure A6-11 shows the paging request at a BAR and Figure A6-12 at the MN.

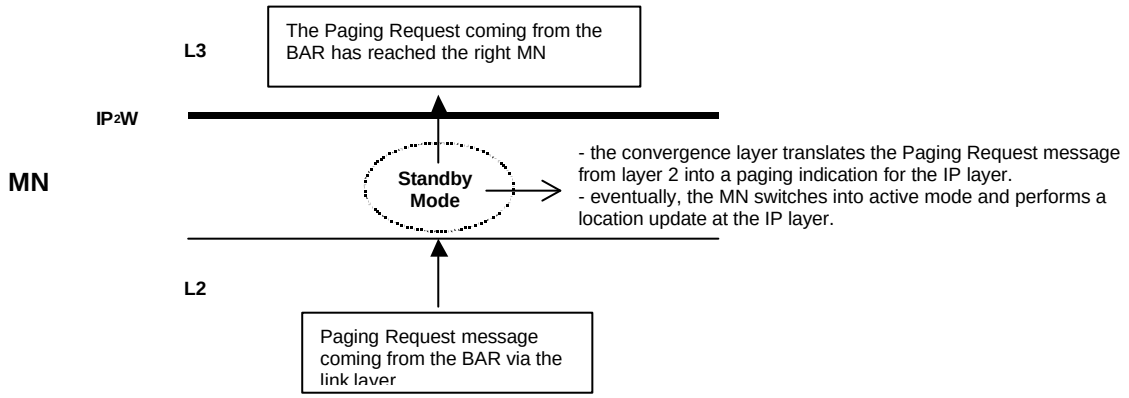


Figure A6-11: Paging Request at the Mobile Nodes

At the MN, the link-layer paging request results in the need of a location update, produced at the network layer.

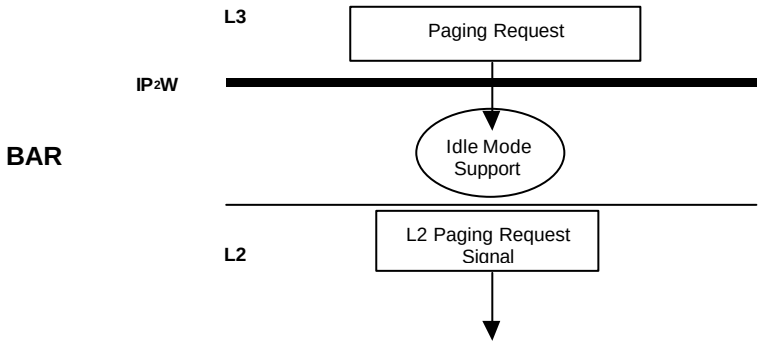


Figure A6-12: Paging Request at the BRAIN Access Router

A6.1.5.3 Example Scenario

Figure A6-13 depicts a scenario that demonstrates link-layer paging support. The MN moves between two BARs that belong to different Paging Areas ("a" and "b").

The MN starts in active mode and it first configures its MN-identifier that it expects to be used in Paging Requests. This configuration is typically performed after the initial registration with the network. The BARs configure their Paging Areas to be able to advertise their Paging Areas in link-layer beacons. When the MN enters the idle mode it (typically) informs the network, waits for a reply, and sets the link-layer in stand-by mode where the MN's receiver only monitors a broadcast channel or a specific paging channel, which conveys information about the current Paging Area and Paging Requests. When the link layer at the MN notices that the Paging Area advertised by a BAR changes, the link layer notifies the upper layers of the change by a Paging Area change notification. This event will wake up the IP stack and typically result in a network-level registration procedure.

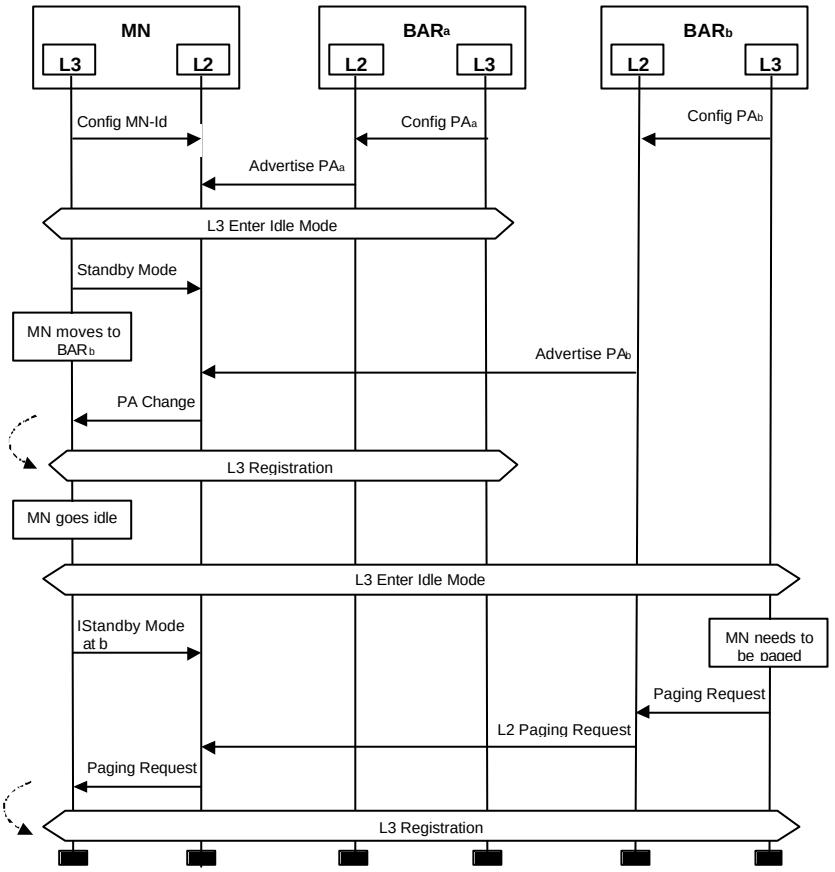


Figure A6-13: Paging Procedures

The scenario continues with a network-layer Paging Request at the BAR. This is forwarded to the link layer through the IP_2W Paging Request primitive, which will trigger a link-layer paging signal from the BAR to the MN. The link-layer at the MN notices that the identifier in the request matches with the configured MN-identifier and notifies the upper layers of a received Paging Request. This will typically result in a network-layer registration procedure, which sets up the routing path in the network for the MN.

When an idle MN wants to send packets, it should initiate a registration procedure with the current BAR. This registration will entail IP packet transmission that will (implicitly) wake up the link-layer.

A6.1.6 Security

A6.1.6.1 Authentication

To perform admission control, the BAN is assumed to operate AAA protocol. Between the BAR and the MN, the authentication mechanism may be performed at the link layer or at the IP level. IP_2W requirements are given below in the case the link layer ensures the authentication of the users.

The IP_2W within the BAR has

?? to indicate whether a Layer 2 admission control is required

?? to transfer credentials for a MN from upper layers to the link layer as the BAR will consult the local authority at the IP level

The IP_2W within an MN has

?? to indicate Layer 3 if authentication is performed at Layer 2

?? to transfer its identity from upper layers to the link layer if required

As it is done with IP layer authentication, a link layer authentication procedure will not cipher credential exchange, since cryptographic exchange can not be performed before the end of the authentication process.

A6.1.6.2 Data Ciphering

Once the MN is authenticated, key exchange procedure like IKE, can be triggered. For example, IKE standard allows ciphering network layer SA (Security Association) exchange as soon as possible. Then the ciphering of data and control packets may be performed at the IP level and optionally at the link layer level. If the link layer level ciphering is independent from the IP level ciphering (depends on the Layer 2 capabilities to generate and exchange its own key to encrypt), no specific function has to be implemented in the IP_2W layer. On the contrary, the MN and the BAR may need to share a securing key that is built and known at the IP level or even at a higher level, for ciphering on the air interface. The Layer 2 then can not be ciphered, while this key is not built and distributed.

In that case, the IP_2W within the BAR has:

?? to indicate whether a Layer 2 ciphering is required or not

?? to provide the encryption key from the IP layer to the link layer, so that encryption key exchange can be performed on the Layer 2 between the BAR and the MN

The IP_2W within an MN has:

?? to indicate to the Layer 3 whether a Layer 2 ciphering is performed or not.

A6.1.7 Configuration

The IP_2W configuration interface enables discovering the capabilities of the wireless link layer and setting parameters for link-layer operation. Accordingly, the configuration interface provides overall management of the functional blocks identified in the control interface and in the data interface. While the individual control functions implement primitives for operational control (e.g., of packet flows or individual packets), the configuration interface is used for managing the general and default modes of operation.

The following sections list the capabilities and the configuration parameters used for enabling or disabling these capabilities and configuring their operation.

A6.1.7.1.1.1 Discovery of link-layer capabilities

The IP_2W must be able to report on the capabilities of the underlying link layer. The link-layer capabilities are categorised into broad capability classes according to the functional blocks of IP_2W interface. If a link-layer implementation claims to support a certain capability it must implement the set of primitives and the underlying functions incorporated in that capability. The capabilities are: -

Address Management

?? support for address autoconfiguration (supersedes DAD)
?? support for optimised address resolution
?? support for promiscuous mode

QoS Control

?? support for int-serv flow specification
?? support for diff-serv mapping
?? predefined QoS contexts

Handover Control

- ?? support for association control
- ?? support for neighbourhood awareness (handover preparations possible)
- ?? support for handover progress monitoring (indications of handover events received)
- ?? support for handover decision control (possible to select handover timing and target)
- ?? support for soft handover

Idle Mode Support

- ?? support for standby mode and paging

Security Management

- ?? support for link-layer encryption
- ?? support for link-layer authentication

Error Control

- ?? support for ARQ

Buffer Management

- ?? support for queue flow control

Header Compression

- ?? support header compression

Multicast Support

- ?? support for link-layer broadcast
- ?? support for link-layer multicast (filtering)

A6.1.7.1.1.2 Configuration of link-layer capabilities

The link layer may be instructed to operate in different modes and with different parameters within the ranges of its capabilities. In addition, the modes of operation and values of fixed and settable parameters can be queried. The possible modes and parameters are: -

Address Management

- ?? enable/disable support for address autoconfiguration
- ?? enable/disable support for optimised address resolution
- ?? enable/disable promiscuous mode

QoS Control

- ?? enable/disable FEC
- ?? enable/disable ARQ

Handover Control

- ?? enable/disable handover monitoring and control modes:
  - ?? neighbourhood awareness
  - ?? handover progress monitoring
  - ?? handover decision control
- ?? get connection status

Idle Mode Support

- ?? enable/disable link-layer paging protocol
- ?? set paging area info at BAR

Security Management

- ?? enable/disable link-layer encryption
- ?? enable/disable link-layer authentication (for admission control)
- ?? get encryption mandatory status (encryption always on)
- ?? set encryption algorithm (DES, RC2, RC4, RC5)

Error Control

- ?? enable/disable ARQ

?? set error control parameters

Buffer Management

Header Compression

?? enable/disable IPv4/IPv6 header compression

Multicast Support

General

?? get access type: connection-oriented or random access
?? get media type (HIPERLAN/2, IEEE 802.11, UMTS , ...)
?? get vendor info
?? get/set network identifier or domain name, ESSID in 802.11
?? get hardware status (ready, initialising, closing, reset, not ready)
?? get link speed (approximate or precise)
?? enable/disable transceiver

A6.1.8 Data Transfer

The Data Interface allows upper layers to send and receive packets over the wireless link. The actual service primitives required for data transfer are fairly limited. The interface support primitives for sending and receiving IPv4 and IPv6 packets and attaching additional interface control information (ICI) to each transmitted/received packet. The ICI includes:

?? source and destination hardware addresses

?? packet length

?? QoS context identifier (optional, acquired through QoS control function)

?? security context identifier (optional, possibly acquired through security management function)

IP packets are mapped to radio resources (e.g. radio channels) by using the interface control information.

When link-layer buffers are becoming exhausted inter-layer flow control is triggered. The flow control indication should be sent before the link layer has to drop packets.

While the Data Interface in itself is very simple, there is a host of behavioural requirements that come from IP layer and upper layers. These do not have a direct impact the interface, but rather affect the implementation of the "user plane procedures" taking care of the actual packet transmission in the link layer. Requirements and recommendations can be identified for the following user plane procedures:

?? **Error Control** refers to mechanisms used to detect and correct transmission errors on the link layer, including error detection, forward error correction, and ARQ. Implications of different error control strategies on IP traffic and on the performance of IP based protocols and applications are considered.

?? **Buffer Management** refers to how buffers are managed in the link layer in conjunction with congestion, QoS, flow control, etc.

?? **QoS Support** refers to scheduling data flows to radio link channels based on QoS parameters.

?? **Segmentation and Reassembly** refers to the procedures of segmenting an IP packet into multiple link layer frames and reassembling them back into an IP packet at the receiving end.

?? **Header Compression** refers to the compression of IP packet headers below the IP_2W interface. Optionally, payload data could also be compressed at this level.

?? **Multicast Transmission** refers to the process of transmitting and receiving packets on multicast channels offered by the underlying wireless technology.

The general requirements for any link layer that is designed for conveying IP packets should be based on the guidelines given by the IETF Performance Implications of Link Characteristics (PILC) working group. These design issues are discussed in the “Advice for Internet Subnetwork Designers” document [A6.14].

A6.2 Interface Specification

This Annex specifies the primitives included in the IP_2W interface.

Unless otherwise noted each primitive is specific to a single interface. Each primitive has an implied interface number parameter. Unless otherwise noted primitive applies for both mobile nodes and access points. The latter refers to a specific scenario found in many link-layers that there is a central-point of control within the link, a so-called access point.

A6.2.1 About Primitive Notation

Four primitive types may be used between different layers:

?? **req** (request), for a higher layer to request service from a lower layer.

?? **cnf** (confirmation), for the layer providing the service to confirm the activity has been completed.

?? **ind** (indication), for the layer providing service to notify the next higher layer of any specific service related activity.

?? **rsp** (response), for a layer to acknowledge receipt of an indication primitive from the next lower layer.

Many of the primitives take the MN's and BAR's identification (MN Id and BAR Id, respectively, or Peer Id) as their parameters. In most cases this identifier will be a Destination Link-layer Address. However, in some occasions it could also be an IP address or a NAI (this needs further study and feedback from the implementers).

A6.2.2 Data Structures and code values

A6.2.2.1 Generic link-layer address (GLLA)

A network interface is assumed to be identifiable by a unique interface identifier within the scope of its use. Network interface is usually configured to use its link-layer address as the interface identifier, so the link-layer address is assumed to possess the same properties. Link-layer address is often referred to as the MAC address but this is a misnomer. No doubt this practice is being influenced by the IEEE 802.x and most notably the widespread Ethernet (IEEE 802.3) standard, which defines a static globally unique link-layer address.

Nevertheless there are link layers, even of the multiple access kind, which do not support static link-layer addresses natively. These are most often found in the wireless links, typically uniqueness only being guaranteed within the link. IP_2W interface must take all kinds of link layers account. See Figure A6-14 for an illustration of the various addresses, identifiers and their relationship. The link-layer usually has an internal presentation for its link-layer address (such as IEEE 802) and then another presentation that is seen by the network layer. This public link-layer address might be directly derived from the internal one or be formed in completely another way.

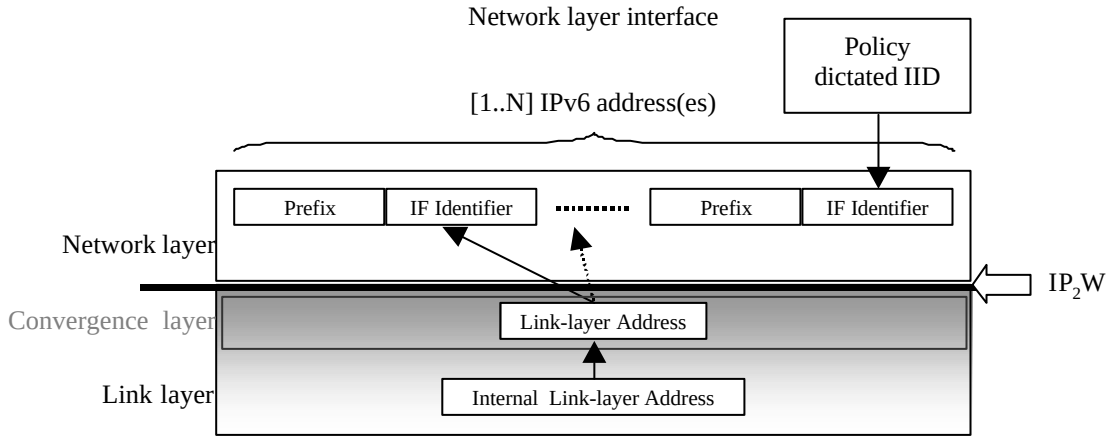


Figure A6-14: The Relation of Various Addresses

To network layer, link-layer addresses should appear as opaque tokens with associated attributes. Table A6-5 details the properties of the generic link-layer address (GLLA):

Property	Explanation
Data	Opaque link-layer address
Length	Length of the link-layer address
Type	Type of the link-layer address – 0 for undefined, 1 for EUI-64
S	Static flag if set, indicates that the link-layer address will not be deprecated. It is globally stable.
U	Unique flag if set, indicates that this address is unique at least within the link.
G	Individual/Group flag to indicate whether this address refers to a hardware multicast address or not.

Table A6-5: Generic Link-layer Address Properties

How to map the true hardware interface address (e.g. MAC address) (if any) to the link-layer address, is a link-specific problem. It is advised that the link-layer addresses should follow the EUI-64 format.

A6.2.2.2 Ethernet Protocol ID

A protocol ID is needed to demultiplex the packet at the network layer. The protocol stack needs to know which protocol handler should take care of the packet. Some of the currently assigned Ethernet protocol identifiers are listed in Table A6-6.

A6.2.2.2.1 Protocol	*A6.2.2.2.2 ode*
IPv4 packet	0x0800
Address Resolution Packet	0x0806
Reverse Address Resolution packet	0x8035
IPv6 packet	0x86dd

Table A6-6: Ethernet Protocol Identifiers

A6.2.3 Address Management Interface

IP_2W should provide a 1) robust, 2) secure and 3) link-independent interface for supporting the network layer address management. Interface should also minimise the need for network layer signalling and maximise the performance if additional link knowledge can be used to aid in the network layer decisions.

A6.2.3.1.1 Requirements

The following requirements for the interface have been identified:

?? The IP_2W interface should accommodate both the shared and point-to-point interface models.

?? IP_2W should provide a static, globally unique link-layer address. If one is not available, the shared interface may support a dynamic hardware address assignment procedure. The procedure must ensure that the assigned link-layer address is unique on the link. Upper layers must be able to query the link-layer address from the interface.

?? The shared interface should support multicast.

A6.2.3.1.2 Discussion

The proposed interface should fulfil all the requirements. Both shared and point-to-point links are supported. In fact the interface tries to abstract them away, so the higher layers need not know the actual type of the link. The defined interface supports both static, globally unique hardware addresses and those, which are dynamically acquired. Higher layers can access this property information in form of the flags described in Table A6-5. The proposed primitives IP2W_JOIN_MULTICAST_GROUP, IP2W_LEAVE_MULTICAST_GROUP and IP2W_MAP_MULTICAST provide the multicast support.

The control interface presents the following functions:

?? Given a raw link-layer address (e.g. acquired from a Link-layer Address option) creates a generic link-layer address structure.

?? Indicates higher layers if link-layer addresses are no longer valid, for example because of a handover.

?? Acquires the link-layer address in link-layer specific ways.

?? Allows the link-layer address be set explicitly.

?? Maps IP multicast addresses to link-layer addresses.

?? Facilitates joining/leaving multicast groups.

Negotiation of IPv4 addresses is considered to be out of the scope of IP_2W.

IP2W_ACQUIRE_LLA {req, cnf }

The request is used by the network layer to acquire the interface link-layer address as a generic link-layer address (GLLA) structure. It can be used as the interface identifiers (IID) or for other purposes. Additional "Random" flag can be specified, if the user wishes not to use a globally known static address.

The confirmation is used by the link layer to inform network layer about the acquired link-layer address.

The optional GLLA as a request parameter can be used to explicitly set the link-layer address to be used. In this case the responsibility of higher layers is to ensure the uniqueness of the address. The request may fail if the link layer does not support the notion of explicitly settable link-layer addresses.

PARAMETER	REQ	CNF	IND	RSP
GLLA	O	A	-	-
Random Flag	O	O	-	-

A	Always present
O	Optional
-	Not applicable

GLLA
A link-layer address, which should be guaranteed to be unique at least within the link. See Table A6-5 for the definition of the generic link-layer address attributes.

Random Flag
This flag indicates that associating the acquired link-layer address to particular hardware or person should not be possible. The flag is passed back to the higher layers if the link-layer is able to provide such an address.

IP2W_QUERY_LLA {req, cnf}

The request is used by the network layer to query link layer about specific properties of a raw link-layer address. Typically this address has been acquired by higher layer mechanisms such as Neighbor Discovery.

The confirmation is used by the link layer to inform the network layer about the properties of the requested address.

PARAMETER	REQ	CNF	IND	RSP
Link-layer Address	A	-	-	-
GLLA	-	A	-	-

A	Always present
-	Not applicable

Link-layer Address
Parameter is a Raw Link-layer Address of which only data and its length are known.

GLLA
Generic link-layer address corresponding to the link-layer address. See Table A6-5 for the definitions of the fields.

IP2W_LLA _DEPRECATION {ind}

This indication is used by the link layer to inform network layer that a particular link-layer address has become deprecated. Typically address deprecation follows handovers in wireless network because the previous interface link-layer address is no longer valid. Higher layers should then acquire a new link-layer address. The lower layers use this primitive to indicate the deprecation of all link-layer addresses,

that is, also link-layer multicast addresses. Higher layers must check the group flag of the address to check whether it refers to a link-layer address or multicast group address.

PARAMETER	REQ	CNF	IND	RSP
Link-layer Address	-	-	A	-

A	Always present
-	Not applicable

Link-layer Address
Higher layers must remove all references to the deprecated link-layer addresses.

IP2W_JOIN_MULTICAST_GROUP {req, cnf}

The request is used by the network layer when joining to a multicast group. It must be noted that this primitive is not optional if IPv6 should be supported, even in the case where link layer does not support multicast or broadcast. The primitive is slightly ugly in that it demands knowledge of the network-layer multicast addressing in lower layer. This ugliness can not be easily avoided because the process of deriving link-layer multicast addresses is really dependent on both the network layer protocol and link layer.

The confirmation is used by the link layer to inform about the outcome of the procedure. The request can fail if there are a limited amount of hardware multicast addresses and if the link-layer for example does not support promiscuous mode. Because of this, the network layer should be careful when ordering the join operations. Mandatory multicast groups such as all-nodes group in IPv6 should be joined before others.

PARAMETER	REQ	CNF	IND	RSP
Network-layer Multicast Address	A	-	-	-
Address Type	A	-	-	-
Link-layer Address	-	A	-	-

A	Always present
-	Not applicable

Network-layer Multicast Address
Primitive expects the address as a structure combining the data portion of the address and its length.

Address Type
Address Type describes the network-layer protocol, typically IPv4 or IPv6. Implementations should support every network-layer protocol that the network stack software supports.

Link-layer Address
Lower layers confirm the success of joining to a multicast group with the link-layer multicast address.

IP2W_LEAVE_MULTICAST_GROUP {req}

This request is used by the network layer to leave a multicast group.

PARAMETER	REQ	CNF	IND	RSP
Link-layer Multicast Address	A	-	-	-

A	Always present
-	Not applicable

Link-layer Multicast Address
This address identifies the link-layer multicast context.

IP2W_MAP_MULTICAST {req, cnf}

The request is used by the network layer to map a multicast address to a link-layer address. The procedure is a distinct operation from joining, because a node need not join to a multicast group, before being able to send datagrams to the group.

The confirmation is used by the link layer to inform the network layer about the link-layer multicast address.

PARAMETER	REQ	CNF	IND	RSP
Network-layer Multicast Address	A	-	-	-

Address Type	A	-	-	-
Link-layer multicast address	-	A	-	-

A	Always present
-	Not applicable

For parameters, see IP2W_JOIN_MULTICAST_GROUP.

The following primitives can be supported optionally:

IP2W_PROPOSE_IID {req, cnf}

The request is used by the network layer to propose an Interface Identifier to the AP. IID could have been formed by taking the link-layer address of the interface, manually or otherwise.

The confirmation informs the network layer about the outcome of the operation.

PARAMETER	REQ	CNF	IND	RSP
Interface Identifier	A	-	-	-
Success Flag	-	A	-	-

A	Always present
-	Not applicable

Interface Identifier
IID to be proposed.

Success Flag
FALSE indicated failure of the operation. MN has to form another IID somehow or use other IP address configuration mechanisms such as DHCP. TRUE indicates success.

IP2W_IID_PROPOSAL {ind, rsp}

The indication is used by the link layer in BAR to inform network layer about new IID proposal. If no duplicate entries are found and other conditions are met, the request by MN can be granted.

The response is used by the network layer to signal the outcome of the operation to the MN.

PARAMETER	REQ	CNF	IND	RSP
Interface Identifier	-	-	A	-
Link-layer Address	-	-	A	-
Success Flag	-	-	-	A

A	Always present
-	Not applicable

Interface Identifier
IID that has been proposed.

Link-layer Address
Link-layer address of the mobile node that proposed the IID.

Success Flag
Status of the proposal.

A6.2.4 Quality-of-Service Control Interface

IP_2W should allow higher layers to take advantage of link layer QoS mechanisms in as generic way as possible. The interface should be link-type agnostic but still allow for precise control of the link layer capabilities.

All flows are presented by an opaque token that is called the QoS context identifier. It is a numerical and unique value that is used by the data interface and the QoS control interface primitives to manage flows on the link layer. It should be unique in the sense that all link-layer flows towards a certain link-layer destination address should have a unique identifier, regardless of the way that they are mapped to physical or logical link layer flows or other QoS mechanisms. This requirement was made to separate the token and the mechanism from each other. The actual link-layer flow can change due to (radio) handovers or other reasons independently of the QoS context ID. Thus, because no remapping of the QoS Context Ids

should be required at a new BAR if link-layer QoS contexts are transferred from an old BAR, the QoS Contexts are identified by the pair (QoS Context Id, MN's link-layer address).

All flows are uni-directional and the interface is fully symmetric. The sender, a MN or a BAR, allocates a flow by using either the IP2W_RESERVE_FLOW.REQ or IP2W_PRIO_FLOW.REQ primitive.

A6.2.4.1 Reservation Based Link Flows

The reservation based service deals with hard agreements of data rates and the transmission delay at the radio interface. Therefore, fixed capacity should be allocated for a specific link-layer flow. The packets belonging to the same flow are delivered in order by the link layer.

IP2W_RESERVE_FLOW {req, cnf }

The request is used by the network layer to reserve capacity on the link. All flows are considered uni-directional and the request primitive is applied in the node that is the sender on the link.

The request should fail if the link layer is incapable of performing access control on the link layer or the corresponding node on the link refuses to accept the reservation. Network layer can either try to change the capacity of an existing QoS context or reserve a completely new one.

The confirmation is used by the link layer to inform about the outcome.

PARAMETER	REQ	CNF	IND	RSP
Destination Link-layer Address	A	A	-	-
QoS Context ID	O	A	-	-
Maximum Packet Size	O	-	-	-
Rate	A	O	-	-
Slack Term	A	O	-	-
Rate Dependent Error Term	-	O	-	-
Per-packet Error Term	-	O	-	-
Loss Rate	O	O	-	-

A	Always present
O	Optional
-	Not applicable

Destination Link-layer Address
Link-layer address of the destination of the flow.

QoS Context ID
An opaque token presenting the QoS context. It should be unique within the life of the QoS context. If the caller adds QoS Context ID, the call requests a change in the reservation of the indicated flow; the other parameters thus indicate what the request change is.

Maximum Packet Size
Optionally, maximum packet size should help link layer to provide better estimations for the slack and error terms. If not specified, then MTU should be used instead.

Rate
The rate is measured in bytes of IP packets per second. This rate is asked to be reserved from the whole capacity. If the link can not provide such throughput then it should report what it can actually provide.

Slack Term
The slack term is measured in microseconds. It signifies the difference between the desired delay and the delay obtained by using a reservation of rate n. This slack term can be utilised by the link layer to reduce its resource reservation for a particular flow.

Rate Dependent Error Term
The rate-dependent error term represents the delay a datagram in the flow might experience due to the rate parameters of the flow. An example of the error term is the need to account for the time taken serialising a datagram broken up into radio cells, with the cells sent at a frequency of 1/rate.

When computing the delay bound, the error term is divided by the reservation rate, the effect of the error term is a function of the transmission rate. Implementers should take care to confirm that their error term values, when divided by various rates, give appropriate results. Delay values that are not dependent on the rate should be incorporated into the value for the per-packet error term.

The rate dependent error term is measured in units of bytes.

Per-packet Error Term
The per-packet error term represents the worst case non-rate-based transit time variation through the link layer. It is generally determined or set at boot or configuration time. An example of the term is a slotted radio link, in which senders are assigned particular slots in a cycle of slots, e.g.. in GSM. Some part of the per-flow delay may be determined by which slots in the cycle are allocated to the flow. In this case, the term would measure the maximum amount of time a flow's data, once ready to be sent, might have to wait for a slot.

The per-packet error term is measured in units of one microsecond.

Loss Rate
The loss rate gives an indication to the link layer about how critical it is to get packets through the link. A low loss rate will force a more persistent retransmission scheme, for example, and possibly changes in the coding. A higher loss rate will let the link layer drop packets more often.

The loss rate is given as a fraction of IP packets that can be lost, for example, 10-2 means, in average, every 100th IP packet can be lost.

IP2W_FLOW_RESV {ind, rsp}

The indication is used by the link layer to inform network layer about pending uni-directional reservations. Network layer can either accept or refuse the reservation based on policy. It must be noted that this primitive must not be mixed with call access control policy in the link layer.

The response is used by the network layer to signal its access control decision.

PARAMETER	REQ	CNF	IND	RSP
Link-layer Address	-	-	A	-
Rate	-	-	A	-
Accept Flag	-	-	-	A

A	Always present
O	Optional
-	Not applicable

Link-layer Address
Link-layer address of the sender.

Rate
The rate in bytes of IP packets per second can aid making the policy decision.

Accept Flag
The decision is signalled by this flag.

IP2W_TEAR_FLOW {req, cnf}

The request is used by the network layer to explicitly tear down a flow presented by a QoS context.

The confirmation is used by the link layer to signal that the flow has been torn down.

PARAMETER	REQ	CNF	IND	RSP
MN Id	A	-	-	-
QoS Context ID	A	-	-	-

A	Always present
O	Optional
-	Not applicable

MN Id
The identification of the MN. This parameter is only applicable at BAR.

QoS Context ID
The context ID of the flow that is to be torn down.

IP2W_FLOW_DEPRECATION {ind}

This indication is used by the link layer to signal that a flow has been deprecated.

PARAMETER	REQ	CNF	IND	RSP
MN Id	-	-	A	-

QoS Context ID	-	-	A	-

A	Always present
O	Optional
-	Not applicable

MN Id
The identification of the MN. This parameter is only applicable at BAR.

QoS Context ID
The context ID of the flow that is deprecated.

IP2W_RESOURCE_VIOLATION {ind}

This indication is used by the link layer to signal that the resources of one specific or all flows have diminished and the indicated flow(s) will be torn down, to be re-allocated again by the IP layer, for example.

PARAMETER	REQ	CNF	IND	RSP
MN Id	-	-	A	-
QoS Context ID	-	-	O	-

A	Always present
O	Optional
-	Not applicable

MN Id
The identification of the MN. This parameter is only applicable at BAR.

QoS Context ID
The context ID of the flow that will be torn down. If the value is not present (equals to zero), then resources of all flows are running out and the higher layers needs to react. This primitive tears down all indicated flows (one specific or all flows) and the higher layer can then re-request resources with some specific logic.

A6.2.4.2 Non-reservation Based Link Flows

The non-reservation based service deals with link flows possibly having a priority and/or a transmission delay bound at the radio interface. If flows with priorities are supported, the link layer serves the flows in priority order following its scheduling algorithm. If flows with priorities are not supported, the link layer serves the flows with different delay bound in an appropriate order (for example using round robin). No fixed capacity should be allocated for such link-layer flows. The packets belonging to the same flow are delivered in order by the link layer.

IP2W_PRIO_FLOW {req, cnf}

The request is used by the network layer to allocate and identify a flow on the link and to assign a specific priority and (reliability related) delay bound to the flow. All flows are considered uni-directional and the request primitive is applied in the node that is the sender on the link. The packets belonging to the same flow are delivered in order by the link layer.

If the network layer does not request any priority nor delay or if the link layer does not support the flow of requested type, the link layer may assign appropriate values to the flow of fail in allocating the flow.

The confirmation is used by the link layer to inform about the outcome.

PARAMETER	REQ	CNF	IND	RSP
Destination Link-layer Address	A	A	-	-
QoS Context ID	O	A	-	-
Priority	A	A	-	-
Delay Bound	O	A	-	-
Reliability Flag	O	-	-	-
Loss Rate	O	O	-	-

A	Always present
O	Optional
-	Not applicable

Destination Link-layer Address

Link-layer address of the destination of the flow.

QoS Context ID
An opaque token identifying the flow and presenting the QoS context. It should be unique within the life of the QoS context. The network layer may request one of the predefined flows by using a QoS Context ID the link layer returned with the IP2W_PREDEFINED_QOS_CLASSES.CNF primitive.

Priority
Priority is a value from 0 to 255, conveniently taking a single octet. If the link layer has no facilities to take the value into account, it may assign the closest supported value that indicates higher priority than the requested value, or it may assign the best effort class, if no priorities are supported.

Delay Bound
Delay bound is used to allow separate handling of packet flows requiring different persistence related delay bound. The Delay bound is considered as the absolute upper bound that a single packet is allowed to wait to be transmitted over the link, excluding the queuing delay. This delay value includes possible retransmissions. Packets exceeding this delay bound should be discarded. Typically lower delay bound indicates the possibility to accept higher packet loss rate. A special value (zero=0) is used to indicate desire for minimum delay.

Reliability Flag
Reliability flag suggests that the higher layer protocols do not care about bit errors in the data portion of the packet. Link-layer framing if any must still be sent untouched but higher layers take care of any corruption in the data portion of the packet. Link-layers that do not have the capability to separate link-layer framing and data portion packet checksum should just ignore this flag.

Loss Rate
The loss rate gives an indication to the link layer about how critical it is to get packets through the link. A low loss rate will force a more persistent retransmission scheme, for example, and possibly changes in the coding. A higher loss rate will let the link layer drop packets more often.

The loss rate is given as a fraction of IP packets that can be lost, for example, 10-2 means, in average, every 100th IP packet can be lost.

IP2W_BUFF_CTRL {req, cnf}

The network layer may use this primitive to get information of the current buffering status information of the QoS Class (flow).

PARAMETER	REQ	CNF	IND	RSP
MN Id	A	-	-	-
QoS Context Id	A	-	-	-
Max Buffer Size	-	A	-	-
Current Buffer Size	-	A	-	-
Quench High Water Mark	-	A	-	-
Quench Low Water Mark	-	A	-	-
Current Min Buffer Size	-	A	-	-

A	Always present
O	Optional
-	Not applicable

MN Id
The identification of the MN. This parameter is only applicable at BAR.

Max Buffer Size
Maximum amount of buffer space allocated for the QoS class associated with the QoS Context ID.

Current Buffer Size
Amount of data currently buffered by the link layer for the QoS class associated with the QoS Context ID.

Quench High Water Mark
Indicates a high water threshold for the data buffered at the link layer for the QoS class associated with the QoS Context ID. When this threshold is exceeded, the link layer will enter the soft stop mode (i.e., start of quench period).

Quench Low Water Mark

Indicates a low water threshold for data buffered at the link layer for the QoS class associated with the QoS Context ID. When the amount of data buffered returns below this threshold, the link layer will leave the soft stop mode (end of quench period).

Current Min Buffer Size
How much data the link layer would like to have buffered for the QoS class associated with the QoS Context ID to be able to efficiently deliver data over the link.

A6.2.5 Association and Handover Control Interface

Typically handover protocols at the network layer are specified in a way that does not rely on link layer signalling between the MN and access routers. However, often any indications of handover events at the link-layer can make a handover protocol more efficient, which adds attractiveness of link layers that support neighbourhood awareness and handover progress monitoring. Independently of these monitoring functions, handover decisions may or may not be controlled via the IP_2W interface.

The IP_2W Handover Control Interface provides an optional set of functions for achieving smooth handovers by supporting network layer handover mechanisms. Handover Control builds on Association Control, which is a mandatory part of IP_2W control functionality. Context transfer via network layer is supported by allowing the upper layers to retrieve link-layer feature contexts at a BAR and to push them down to the link-layer at another BAR.

The naming of the handover primitives suggests a connection-oriented link. When a pure random-access link is used, the notion of "attachment" can be interpreted as "providing link-layer access".

A6.2.5.1 Handover Phases

The handover-phasing model is illustrated in Figure A6-15. The figure shows the relevant commands and events that are relevant in various handover phases:

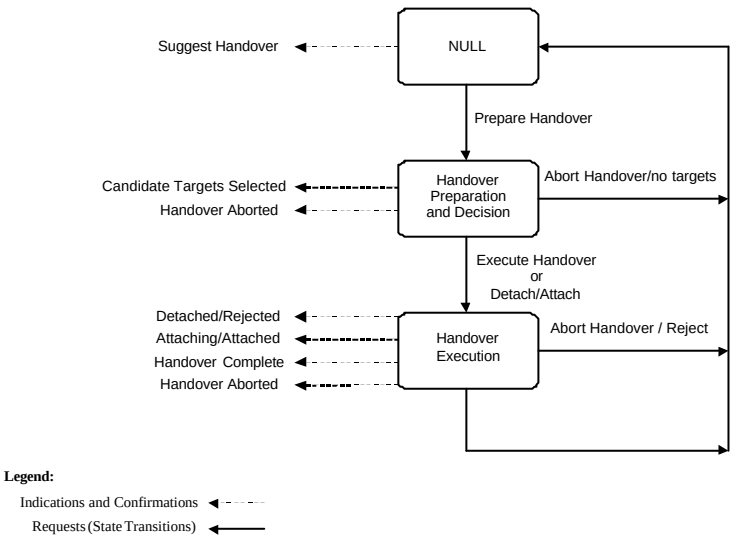


Figure A6-15: Handover Phases

The phase diagram is primarily seen from the MN's point of view. However, it is also applicable to BAR. Any phase transitions shown in the figure may happen either after following a command or spontaneously. Note also that the figure does not show all possible transitions. For example, the Handover Preparation phase is not a prerequisite for the Handover Execution phase if NAR[86] can be known by other means than through the IP_2W candidate BAR selection procedure. Similarly, Handover Preparation may be started without a preceding Handover Suggest indication. For example, the network layer may recognise an idle period in IP packet transmission, which may be used as a trigger for starting radio measurements.

[86] Note that the granularity of identification of point of attachment is an interface of an AR and not the whole AR node if it has multiple wireless interfaces.

A6.2.5.2 Association Control

IP2W_NODE_ATTACH {req, cnf, ind, rsp}

The request is used to attach the MN to a new BAR.

The confirmation confirms a successful attach.

The indication is sent by the link layer to inform the network layer in the MN or BAR that a MN has performed a handover and acquired a link-layer address for itself. The event

?? may be spontaneous (e.g., the MN has powered up at the new BAR),

?? may be initiated by an IP2W_NODE_ATTACH or IP2W_HANDOVER_EXECUTE request at the peer node.

The response accepts the attachment of a peer node.

A BAR may not want to admit the attaching MN. Alternatively, the MN may not want to get served by the new BAR (in a network-controlled handover). Then the BAR (or the MN) may reject the attachment by IP2W_NODE_REJECT request. It should be configurable whether the link-layer automatically accepts an attaching node or whether it must wait for an IP2W_NODE_ATTACH response from the upper layers.

PARAMETER	REQ	CNF	IND	RSP
MN Id	A	A	A	A
BAR Id	A	A	A	A
Reason	A	A	A	-
Timeout	A	A	A	-
Link-layer Address	A	-	A	-
QoS Context Id	-	A	A	-

A	Always present
O	Optional
-	Not applicable

MN Id
The identification of the MN. This parameter is only applicable at BAR.

BAR Id
The identification of the new BAR. This parameter is mainly applicable at MN. The parameter may not be valid unless it has appeared in the target list of the IP2W_HANDOVER_TARGETS_SELECTED primitive.

Reason
This parameter should be conveyed to the peer for information. Possible reasons are - 1) the peer has performed an (initial) attach procedure or 2) handover has been executed.

Timeout
The Timeout parameter specifies the maximum time in seconds that is allowed for attaching.

Link-layer Address
Link-layer address of the mobile node (may be the same as MN Id).

QoSContext ID
The default context ID used for all traffic at attachment time.

IP2W_NODE_DETACH {req, cnf, ind}

The request is used to detach the MN from the currently serving BAR.

The confirmation confirms a successful detachment or rejection (see, IP2W_HANDOVER_REJECT).

The indication informs that the MN has detached from its current BAR. The indication

?? may be spontaneous (e.g., the connectivity between the MN and the currently serving BAR is suddenly lost), or it

?? may indicate the first phase of a handover that may have been initiated by IP2W_HANDOVER_EXECUTE request.

If "Retain Addresses" flag is specified, the node wants to retain its interface identifiers for a possible later re-attachment.

PARAMETER	REQ	CNF	IND	RSP
MN Id	A	A	A	-
BAR Id	A	A	A	-
Reason	A	-	A	-
Timeout	A	-	A	-
Retain Addresses	O	O	O	-

A	Always present
O	Optional
-	Not applicable

MN Id
The identification of the MN. This parameter is only applicable at BAR.

BAR Id
The identification of the BAR. This parameter is mainly applicable at MN.

Reason
This parameter indicates the reason for detachment. Possible reasons are - 1) lost of connection (typically after a timeout), 2) explicit detach to release resources (either local confirmation or indication at the peer), or 3) handover executed.

Timeout
The Timeout parameter specifies the maximum time in seconds that is allowed for detaching.

Retain Addresses
Flag that signifies that allocated interface identifiers should be retained.

IP2W_NODE_REJECT {rsp, ind}

The response is used to reject an attachment that has been announced in IP2W_NODE_ATTACH indication. This request is normally given by a node that has not initiated the handover execution or an initial attachment.

The indication notifies of the rejection of an attachment procedure that has been initiated either with the IP2W_NODE_ATTACH request or with the IP2W_HANDOVER_EXECUTE primitive. The indication may be preceded or followed by an IP2W_NODE_ATTACH indication, which indicates that the MN is returning back to the old BAR. This indication may also be accompanied with the IP2W_HANDOVER_ABORTED indication.

PARAMETER	REQ	CNF	IND	RSP
MN Id	-	-	A	A
BAR Id	-	-	A	A
Reason	-	-	A	A

A	Always present
O	Optional
-	Not applicable

MN Id
The identification of the MN. This parameter is only applicable at BAR.

BAR Id
The identification of the intended new BAR. This parameter is mainly applicable at MN.

Reason
The reason for rejection.

A6.2.5.3 Handover Control

IP2W_HANDOVER_SUGGEST {ind}

This indication is sent by the link layer to inform the upper layers of the need for a handover. The indication may be due to a degradation of radio signal quality or emergence, new neighbouring transmitters, or pre-emption, for example.

PARAMETER	REQ	CNF	IND	RSP
MN Id	-	-	A	-
Reason	-	-	O	-

A	Always present
O	Optional
-	Not applicable

MN Id
The identification of the MN. This parameter is only applicable at BAR.

Reason
This parameter indicates the reason for a need for a handover. It may also indicate the urgency for starting handover preparations.

IP2W_HANDOVER_PREPARE {req, cnf}

The request is used to initiate the handover preparation phase. The link layer should start measures to find the most appropriate set of BARs to which the MN can attach. The mechanisms for determining this set are internal to the link layer and they may require consultation with radio resource management.

The confirmation is used to inform of the available BARs to which the MN may handover. The list of targets consists of structures that indicate the identification of the target BAR and its preference. The BARs are listed in decreasing preference value.

PARAMETER	REQ	CNF	IND	RSP
MN Id	A	A	-	-
Method	O	-	-	-
Timeout	A	-	-	-
BAR Target List	-	A	-	-

A	Always present
O	Optional
-	Not applicable

MN Id
The identification of the MN. This parameter is only applicable at BAR.

Method
The Method parameter may give hints of accessibility of neighbouring access points and the criticality and urgency of handover. This parameter may be opaque to the network layer, and it is not needed if the link layer is able to get this information without help from the upper layers.

Timeout
The Timeout parameter specifies the maximum time in seconds that is allowed for determining the set of candidate BARs for handover targets. For example, the link layer must not perform radio signal measurements for a longer time that is indicated by the Timeout value.

BAR Target List
The list of candidate target BARs (CARs).

IP2W_HANDOVER_EXECUTE {req, cnf}

The request is used to start the handover execution phase. The list of possible target BARs is given.

The confirmation is used to inform of the completion of a handover procedure that has been started with the IP2W_HANDOVER_EXECUTE request. If a backward handover is initiated by OAR, this confirmation may or may not be received at the BAR. Instead, the BAR may only receive an IP2W_NODE_DETACH indication.

PARAMETER	REQ	CNF	IND	RSP
MN Id	A	A	-	-
BAR Target List	A	A	-	-
Timeout	A	-	-	-
BAR Source	-	A	-	-

A	Always present
O	Optional
-	Not applicable

MN Id
The identification of the MN. This parameter is only applicable at BAR.

BAR Target List
The Target List parameter gives the list of handover targets. The list contains the identification of the candidate BARs in the order of decreasing priority. In the confirmation this list only contains a single BAR (i.e., NAR).

Timeout
The Timeout parameter specifies the maximum time in seconds that is allowed for executing the handover.

BAR Source
The identification of the source BAR (OAR).

IP2W_HANDOVER_ABORT {req, cnf, ind}

The request is used to stop the handover execution phase started by the IP2W_HANDOVER_EXECUTE request. If the MN has already detached from OAR it should be reattached to the OAR. If the handover to NAR is already complete the link must not try to return the MN to OAR.

The confirmation confirms the abortion.

The indication is used to inform of the abortion of a handover procedure that has been started with the IP2W_HANDOVER_EXECUTE primitive. If a backward handover has been initiated by a BAR, this indication may or may not be received at the BAR if the MN has already detached from the BAR. This indication may be preceded or followed by an IP2W_NODE_ATTACH indication, which indicates returning back to OAR.

PARAMETER	REQ	CNF	IND	RSP
MN Id	A	A	A	-
Reason	A	A	A	-
Timeout	A	-	-	-
BAR Source	-	A	A	-
BAR Target	-	A	A	-
BAR Current	-	A	A	-

A	Always present
O	Optional
-	Not applicable

MN Id
The identification of the MN. This parameter is only applicable at BAR.

Reason
This parameter indicates the reason for handover abortion.

Timeout
The Timeout parameter specifies the maximum time in seconds that is allowed for aborting the handover.

BAR Source
The identification of the source BAR (OAR).

BAR Target
The identification of the intended target BAR (NAR).

BAR Current
The identification of the current BAR (must be either OAR or NAR).

A6.2.5.4 Context Transfer

Context transfer primitives do not initiate transfer signalling but they enable accessing the MN's state information that resides at the link layer. The state information is assumed to consist of data elements associated with various protocol mechanisms. A group of mechanisms constitute a functionality that is called a feature.

IP2W_GET_MN_CONTEXT {req, cnf}

This primitive is used for retrieving a MN's link-layer context information at BAR. The parameters for this primitive are still for further study.

PARAMETER	REQ	CNF	IND	RSP
MN Id	A	-	-	-
Feature Context Ids	A	-	-	-
Feature Contexts	-	A	-	-

A	Always present
O	Optional
-	Not applicable

MN Id
The identification of the MN.

Feature Context Ids
The identifications of the feature contexts that are to be retrieved. A single identifier may refer to several context instances that belong to different micro-flows, for example.

Feature Contexts
The list of feature contexts.

IP2W_SET_MN_CONTEXT {req, cnf}

This primitive is used for setting the MN's link-layer context information at BAR. The parameters for this primitive are still for further study.

PARAMETER	REQ	CNF	IND	RSP
MN Id	A	-	-	-
Feature Contexts	A	-	-	-
Return Value	-	A	-	-

A	Always present
O	Optional
-	Not applicable

MN Id
The identification of the MN.

Feature Contexts
The list of feature contexts that are to be set.

Return Value
This value indicates whether the context set-up succeeded or failed.

A6.2.6 Idle Mode Interface

IP2W_PAGING_MN_INFO {req}

The request is sent by the network layer in the MN to provide the link layer with the MN identifier, that is used to detect a layer 2 paging request addressed to the MN. This identifier is provided following a regular LOGIN procedure.

PARAMETER	REQ	CNF	IND	RSP
MN Paging Id	A	-	-	-

A	Always present
O	Optional
-	Not applicable

MN Paging Id
The identification of the MN which will be received in the link-layer Paging Request message.

IP2W_STANDBY_MODE {req}

The request is sent by the network layer to inform the link layer in the MN that the MN becomes idle and that the standby mode can be triggered. The event that triggers the idle mode is implementation specific (e.g. timer expiration in absence of upstream data packets, absence of TCP connection…)

PARAMETER		REQ	CNF	IND	RSP
A	Always present				
O	Optional				
-	Not applicable				

IP2W_PAGING_REQUEST {req, ind}

The request is sent by the network layer to inform the link layer in the BAR that it receives a paging request for the MN identified with MN Paging Id (or the BAR itself needs to send an IP packet to the MN) and that it has to page this MN on the link layer.

The indication is sent by the link layer to inform the network layer in the MN that the MN receives a paging request. On receipt of this primitive, the network layer has to trigger the relevant network layer procedure, depending on the MM protocol. In the case of the BRAIN micro-mobility protocol, the MN has to initiate a regular handover (HOFF message) to update its routing entries and to eventually initiate a context transfer between the old BAR and the new BAR if the MN performed a handover when it was idle.

PARAMETER		REQ	CNF	IND	RSP
MN Paging Id		A	-	-	-
BAR Id		-	-	A	-
A	Always present				
O	Optional				
-	Not applicable				

MN Paging Id
The identification of the MN to be inserted in the link-layer Paging Request message.

BAR Id
The identification of the. This parameter is only mandatory in the MN to detect a move for exa mple.

IP2W_PA_CHANGE {ind}

The indication is sent by the link layer to inform the network layer in the MN that the MN changes its paging area (paging information received by the MN changed). On receipt of this primitive, the network layer has to trigger the relevant network layer procedure, i.e. the MN has to initiate a regular handover (HOFF message) to update its routing entries and to eventually initiate a context transfer between the old BAR and the new BAR.

PARAMETER		REQ	CNF	IND	RSP
New PA Id				O	
A	Always present				
O	Optional				
-	Not applicable				

New PA Id
The parameter indicates the new identifier of the paging area in which the MN resides. The New PA Id may be optionally a list of PA Ids to allow paging area overlapping

A6.2.7 Security Management Interface

The security interface primitives allow setting up security contexts at the link layer for encryption and performing a challenge/response handshake for authentication.

IP2W_KEY_REQUEST {ind}

This primitive is used by the link layer to request for keys that are needed for communicating with a peer node.

PARAMETER	REQ	CNF	IND	RSP
Peer Id	A	-	-	-
Key Types	A	-	-	-

A	Always present
O	Optional
-	Not applicable

Peer Id
The identification of the peer node. This may be either an MN or a BAR.

Key Types
The types of the requested keys. The link layer may request for a PIN Code, authentication key or an encryption key, or all of them.

IP2W_SET_KEYS {req, cnf}

This primitive is used for setting up the authentication and/or data encryption keys. Key generation is link-layer dependent (for example, an encryption key may be generated from an authentication key). This primitive takes a PIN-code, link key, and encryption key as optional parameters that can be used as keys or factors in key generation.

The request may be may sent spontaneously or as a response to the IP2W_KEY_REQUEST indication.

The confirmation may return a context identifier for the security context that is used for packet transmission.

PARAMETER	REQ	CNF	IND	RSP
Peer Id	A	-	-	-
PIN	O	-	-	-
Link Key	O	-	-	-
Encryption Key	O	-		
Security Context ID	-	O		

A	Always present
O	Optional
-	Not applicable

Peer Id
The identification of the peer node. This may be either an MN Id or a BAR Id.

PIN
A variable length PIN code that can be used for generating keys for authentication and encryption.

Link Key
A key that can be used for authentication, generating authentication keys, and/or generating keys for encryption

Encryption Key
A key that can be used for encryption.

Security Context ID
The identification of the security context, which is used as one element of the ICI when transmitting data packets. Thus, this context relates to data encryption (rather than authentication).

IP2W_AUTHENTICATION_COMPLETE {ind}

This primitive is sent by the link layer to inform about the completion of the authentication phase.

PARAMETER	REQ	CNF	IND	RSP
MN Id	-	-	A	-
Event Code	-	-	A	-
Error Code	-	-	A	

A	Always present
O	Optional
-	Not applicable

MN Id
The identification of the MN. This parameter is only applicable at BAR.

Event Code

An event code that indicates reception of a valid Response to a Challenge, an acknowledgement of a sent Response, and/or completion of a two-way authentication handshake.

Error Code
An error code that indicates whether authentication succeeded or failed. Possible codes for errors are “bad authentication”, “missing challenge”, “stale challenge”, or “timeout”, for example.

A6.2.8 Configuration Interface

The configuration primitives allow retrieving link-layer capabilities and getting/setting modes of operation and operational parameters. Parameters that pertain to the whole interface instance are accessed through the configuration interface while flow-wise and MN-wise (at the BAR) parameters are accessed through other control functions that are specific to certain functional blocks.

IP2W_CAPABILITY {req, cnf}

Test whether the link-layer supports a particular capability defined in the IP_2W interface.

PARAMETER	REQ	CNF	IND	RSP
Capability	A	-	-	-
Supported Flag	-	A	-	-
Capability Info	-	-	-	-

A	Always present
O	Optional
-	Not applicable

Capability
IP_2W capability identifier.

The following capability identifiers have been defined:

IP2W_CAP_ADDRESS_AUTO_CONF	support for address autoconfiguration
IP2W_CAP_ADDRESS_RESOLUTION	support for optimised address resolution
IP2W_CAP_PROMISCUOUS_MODE	support for promiscuous mode
IP2W_CAP_INT_SERV	support for int-serv flow specifications
IP2W_CAP_DIFF_SERV	support for diff-serv mapping
IP2W_CAP_ASSOCIATION_CONTROL	support for association control
IP2W_CAP_HANDOVER_NA	support for neighbourhood awareness
IP2W_CAP_HANDOVER_PM	support for handover progress monitoring
IP2W_CAP_HANDOVER_DC	support for handover decision control
IP2W_CAP_HANDOVER_SOFT	support for soft handover
IP2W_CAP_PAGING	support for link-layer paging
IP2W_CAP_POWER_MANAGEMENT	support for power mgmt and standby mode
IP2W_CAP_ENCRYPTION	support for link-layer encryption
IP2W_CAP_AUTHENTICATION	support for link-layer authentication
IP2W_CAP_ARQ	support for ARQ
IP2W_CAP_FLOW_CONTROL	support for queue flow control
IP2W_CAP_HEADER_COMPRESSION	support header compression
IP2W_CAP_BROADCAST	support for link-layer broadcast
IP2W_CAP_MULTICAST	support for link-layer multicast
IP2W_CAP_FORWARD_CAPACITY	support query of forward link capacity
IP2W_CAP_BACKWARD_CAPACITY	support query of backward link capacity

Supported Flag

True or False. Indicates, whether the link layer supports the identified capability.

Capability Info
This parameter supplements the binary-valued supported flag by further specifying the capability. For example, it may indicate what header compression modes are supported (e.g., IPv4/6, IPv4/6 UDP/RTP, IPv4/6 TCP).

IP2W_QUERY_QOS_CAPABILITIES {req, cnf}

The network layer uses this primitive to acquire information from the link layer about supported QoS capabilities, including predefined link layer QoS classes (flows) and their properties (parameters).

PARAMETER	REQ	CNF	IND	RSP
Resv Flows	-	A	-	-
Prior Flow Classes	-	A	-	-
Dynamic Prior Flows Flag	-	A	-	-

A	Always present
O	Optional
-	Not applicable

Resv Flows
A numeric value indicating whether the link layer supports reservation of capacity for link-layer flows. If the returned value is zero, reservation-based flows are not supported. A non-zero value indicates the maximum number for the link-layer flows that the network layer may request using the IP2W_RESERVE_FLOW primitive. Note that the network layer may not be able to request the maximum number of flows as the number of flows the link is actually able to serve depends heavily on the capacity reserved for each allocated flow.

Prior Flow Classes
The confirmation returns a list of tuples, one per predefined priority class/flow. A tuple includes the following entries:

?? QoS Context ID: a predefined QoS Context ID that identifies the flow,
?? Priority: the priority of the flow,
?? Delay bound: the delay bound associated with the flow.

Dynamic Prior Flows Flag
If this flag is set, it indicates that the link layer support dynamically created non-reservation based link flows. That is, the network layer may use IP2W_PRIO_FLOW.REQ primitive to request a new link flow with parameters different from those listed in the set of predefined priority flows.

IP2W_SET_PARAMETER {req, cnf}

Set a configurable parameter to value.

PARAMETER	REQ	CNF	IND	RSP
Parameter	A	-	-	-
Value	A	-	-	-
Error	-	A	-	-

A	Always present
O	Optional
-	Not applicable

Parameter
IP_2W parameter identifier. The parameters listed in the table below have been defined. Some of the parameters are read-only, while some parameter values are also changeable (i.e., the “get parameter”-operation is applicable to all parameters). Some of the parameters are toggles that enable or disable a piece of functionality (capability).

IP2W_ENA_ADDRESS_AUTOCONF	enable/disable support for address autoconfiguration
IP2W_ENA_ADDRESS_RESOLUTION	enable/disable support for optimised address resolution
IP2W_ENA_PROMISCUOUS	enable/disable promiscuous mode
IP2W_ENA_HANDOVER_NA	enable/disable neighbourhood awareness
IP2W_ENA_HANDOVER_PM	enable/disable handover progress monitoring

IP2W_ENA_HANDOVER_DC	enable/disable handover decision control
IP2W_PAR_CONNECTION_STATUS	get connection status
IP2W_ENA_PAGING	enable/disable link-layer paging protocol
IP2W_PAR_PAGING_BAR_INFO	set paging area identifier(s) + optionally other info at BAR
IP2W_ENA_ENCRYPTION	enable/disable link-layer encryption
IP2W_ENA_AUTHENTICATION	enable/disable link-layer authentication
IP2W_PAR_ENCRYPTION_MANDATE	get encryption mandatory flag
IP2W_PAR_ENCRYPTION_ALGO	set encryption algorithm
IP2W_PAR_SECRET_KEY	set secret key
IP2W_ENA_ARQ	enable/disable ARQ
IP2W_PAR_ARQ_MAX_RETRY	set error control parameters
IP2W_ENA_HEADER_COMPRESSION	enable/disable IPv4/IPv6 header compression
IP2W_PAR_ACCESS_TYPE	get access type: connection-oriented or random access
IP2W_PAR_MEDIA_TYPE	get media type
IP2W_PAR_VENDOR_INFO	get vendor info
IP2W_PAR_NETWORK_ID	get/set network identifier or domain name
IP2W_PAR_HARDWARE_STATUS	get hardware status
IP2W_PAR_LINK_SPEED	get link speed
IP2W_PAR_TRANSCEIVER	enable/disable transceiver

Value
Parameter value.

Error
Error return code. Success=0, Not Supported=1.

IP2W_GET_PARAMETER {req, cnf}

Get the value of a configuration parameter.

PARAMETER	REQ	CNF	IND	RSP
Parameter	A	-	-	-
Value	-	A	-	-

A	Always present
O	Optional
-	Not applicable

Parameter
IP_2W parameter identifier (see IP2W_SET_PARAMETER).

Value
Parameter value.

IP2W_GET_CAPACITY {req, cnf}

This primitive is used for enquiring the capacity of the link.

PARAMETER	REQ	CNF	IND	RSP
Destination Link-Layer Address	A	A	-	-
Direction	A	A	-	-
Capacity Type	A	A	-	-
Available Capacity	-	A	-	-

A	Always present
O	Optional
-	Not applicable

Destination Link-layer Address
Link-layer address of the peer node.

Direction

Indicates which direction the query should return the capacity for. Values '0' for forward link and '1' for backward link. A link layer may be able to support a query about the sudden effective forward and backward capacity, one direction or the feature may not be available.

Capacity Type
Defines whether the query is for the links nominal capacity or the effective capacity at the time of the call. Value '0' is for nominal capacity and value '1' is for the effective capacity.

Available Capacity
The value is the available bandwidth, expressed in bits per second. If the query was about the effective capacity, the answer provides the available capacity, part of which is may be in use. A negative effective capacity indicates that a QoS violation is imminent, that a resource outage is taking place. This can indicate the caller to reduce the bandwidth usage of some flow.

Note that at least the nominal capacity of the forward link must be supplied to the caller. If the link layer address in missing, the call must return the overall capacity of the link behind that network interface.

A6.2.9 Data Interface

IP2W_SEND_PACKET {req, cnf}

The request is used by the network layer to send packets on the link. The parameters consist of the packet data and interface control information.

The confirmation is used by the link layer to inform network layer about the outcome of the operation. It confirms that the packet seems sane and can be queued not that it has been sent), or indication of inability to accept the packet for delivery. If the link layer buffers have no more space or there is any other reason why the packet can not be accepted for delivery, the confirmation should indicate this (see Dropped Flag).

PARAMETER	REQ	CNF	IND	RSP
Packet Data	A	-	-	-
ICI				
Source Link-layer Address	A	A	-	-
Destination Link-layer Address	A	A	-	-
Packet Length	A	A	-	-
Protocol	A	A	-	-
QoS Context ID	O	-	-	-
Security Context ID	O	-	-	-
Compressed Flag	A	-	-	-
Dropped Flag	-	A	-	-
Quench Flag	-	O	-	-
Outage Flag	-	O	-	-
Buffer Size	-	O	-	-

A	Always present
O	Optional
-	Not applicable

Packet Data
Packet data includes payload and any higher layer headers. The length of this data should not exceed the link Maximum Transmission Unit (MTU). As far as the link-layer is concerned, the contents of the packet should not matter.

Interface Control Information (ICI)
ICI holds per-packet control information. The fields from Source Link-layer Address to Compressed Flag below are parts of the ICI.

Source Link-layer Address
Explicitly specifies the source link-layer address to be used. It should be one of the link-layer addresses acquired with IP2W_ACQUIRE_LLA. The link layer may send the packet with the specified link-layer address if it is able to do so without too much overhead. In particular, if the link-layer framing does not include a source link-layer address field and the specified address is not a valid one, the link-layer should refuse to send the packet.

Destination Link-layer Address
Specifies the destination link-layer address. Link-layer can refuse to send the packet if the address is not valid.

Packet Length
This parameter specifies the packet length in octets (bytes). It must include all of the payload and higher layer headers. If the length exceeds the link MTU, the link must refuse to send the packet.

Protocol
Protocol, a 16-bit value, refers to the network-layer protocol that should handle the packet. It must be sent and received unmodified by the lower layers. See Table A6-5 for currently assigned protocol identifiers.

QoS Context ID
QoS context identifier is an opaque token acquired with IP2W_RESERVE_FLOW or IP2W_PRIO_FLOW to identify a flow the packet belongs to at the link layer. Higher layers do not assume anything of its format. Link-layers may use it as they wish.

Security Context ID
Security Context ID defines the security context for data encryption, if available and enabled.

Compressed Flag
Compressed flag suggests that the payload portion of this packet is already compressed. No link-layer compression should be performed.

Dropped Flag
If the Dropped flag is set it indicates that the link layer was unable to accept the data packet from the network layer due to lack of buffer space. This always indicates hard stop condition (traditional XOFF) and network layer should not try to send a new packet on that flow (with the same QoS Context ID) until IP2W_FLOW_CONTINUE primitive is invoked by the link layer indicating it is able to accept new packets again with the same QoS Context ID.

Quench Flag
If this flag is set it indicates a start of a quench period on the flow with the given QoS Context ID. This is a soft stop condition with which the link layer indicates that it has reached its high water mark and may soon run out of buffer space if the network layer continues sending new packets at too fast rate. Hence, it is an advise to the network layer to stop sending new packets until the link layer invokes the IP2W_FLOW_CONTINUE primitive with the soft continue flag (Continue flag) set indicating the end of a quench period on the given flow.

Outage Flag
Outage flag is an extra hint to higher layers indicating a possible communication problem between the two end-points - an access point or a mobile node. This problem may be due to MN being out-of-coverage or suspected of experiencing outage. It needs not be an accurate prediction on the access point side.

Buffer Size
Informs the network layer of the amount of data (in octets) buffered at the link layer on the data flow associated with the given QoS Context ID. The returned value indicates the situation after enqueueing the data packet. Network layer scheduling algorithms may use this information to perform more effective active queue management and scheduling.

IP2W_PACKET _RECEIVE {ind}

Indication used by the link layer to inform network layer about reception of a new packet.

PARAMETER	REQ	CNF	IND	RSP
Packet Data	-	-	O	-
Source Link-layer Address	-	-	A	-
Destination Link-layer Address	-	-	A	-
Packet Length	-	-	A	-
Protocol	-	-	A	-
Security Context ID	-	-	O	-
Reliability Flag	-	-	O	-

A	Always present
O	Optional
-	Not applicable

Packet Data
Packet data includes payload and any higher layer headers.

Source Link-layer Address

Specifies the source link-layer address.

Destination Link-layer Address
Specifies the destination link-layer address. Unless promiscuous mode is configured, it must be one of the link-layer addresses acquired with IP2W_ACQUIRE_LLA or IP2W_JOIN_MULTICAST_GROUP.

Packet Length
This parameter specifies the packet length in octets (bytes).

Protocol
Protocol, a 16-bit value, refers to the network-layer protocol that should handle the packet. It must be sent and received unmodified by the lower layers. See Table A6-5 for currently assigned protocol identifiers.

Security Context ID
The identification of the security context used for the received packet.

Reliability Flag
Reliability flag indicates that the packet failed the integrity check for data portion of the packet. The flag refers to the capability to send packets with errors (see IP2W_SEND_PACKET). No other packets should be let through corrupted than those suggested by the sender with the dirty flag. At receiver this flag is an extra hint that the packet might be corrupted.

IP2W_FLOW_CONTINUE {ind}

This indication is used by the link layer to release the transmission flow from the network layer. In case of QoS enhanced flows, more advanced flow control information needs to be communicated with the link layers in addition to the traditional XON/XOFF flow control. Network layer scheduling algorithms need to know the exact amount of data buffered currently at the link layer to perform effective active queue management and scheduling. The link layer indicates its inability to accept new data packets from the network layer by the IP2W_PACKET_SEND confirmation. IP2W_FLOW_CONTINUE primitive should be invoked, when the link layer is again able to accept new packets for delivery. This primitive always indicates release of the hard stop condition (traditional XOFF). In addition to that a soft continue flag (Continue flag) may be used to indicate the end of a quench period that was earlier triggered with IP2W_PACKET_SEND confirmation.

PARAMETER	REQ	CNF	IND	RSP
MN Id	-	-	A	-
QoS Context ID	-	-	O	-
Continue Flag	-	-	O	-
Outage Flag	-	-	O	-
Buffer Size	-	-	O	-

A	Always present
O	Optional
-	Not applicable

MN Id
The identification of the MN (or MN's peer node).

QoS Context ID
QoS context identifier is an opaque token acquired with IP2W_RESERVE_FLOW or IP2W_PRIO_FLOW primitive. The flow info should be reported separately for each link-layer QoS context.

Continue Flag
A flow control flag indicates the end of the quench period on the given flow triggered with an earlier IP2W_PACKET_SEND confirmation. The link layer sets this flag on, when it has been able to drain the data packets on the flow so that the amount of data buffered is now below its low water mark.

Outage Flag
Outage flag is extra hint to higher layers indicating end of a possible communication problem between the two end-points - an access point or a mobile node. The beginning of such communication problem is earlier indicated with an IP2W_PACKET_SEND confirmation. This problem may be due to MN being out-of-coverage or suspected of experiencing outage. It needs not be an accurate prediction on the access point side.

Buffer Size

Informs the network layer of the amount of currently buffered data (in octets) at the link layer associated with the QoS Context ID. Network layer scheduling algorithms may use this information to perform more effective active queue management and scheduling.

A6.3 References

[A6.1] Johnson, C. Perkins, “Mobility Support in IPv6”, Internet draft (work in progress), draft-ietf-mobileip-ipv6-12.txt, April 2000.

[A6.2] T. Narten, E. Nordmark, W. Simpson, “Neighbor Discovery for IP Version 6 (IPv6)”, IETF RFC 2461, December 1998.

[A6.3] R. Hinden, S. Deering, “Internet Protocol Version (IPv6) Addressing Architecture”, IETF RFC 2373, July 1998.

[A6.4] S. Thomson, T. Narten, “IPv6 Stateless Address Autoconfiguration”, IETF RFC 2462, December 1998.

[A6.5] S. Blake, et. al., “An Architecture for Differentiated Services”. IETF RFC 2475, Dec. 1998.

[A6.6] R. Braden, D. Clark, S. Shenker, “Integrated Services in the Internet Architecture: an Overview “, IETF RFC 1633, June1994.

[A6.7] R. Braden, et. al., “Resource ReSerVation Protocol (RSVP) -- Version 1, Functional Specification”, IETF RFC 2205, September 1997.

[A6.8] J. Wroclawski, “ The Use of RSVP with IETF Integrated Services”, IETF RFC 2210, September 1997.

[A6.9] R. Yavatkar, et. al., “SBM (Subnet Band-width Manager”, IETF RFC 2814, May 2000.

[A6.10] K. Ramakrishnan, S. Floyd, “A Proposal to Add Explicit Congestion Notification (ECN) to IP”, IETF RFC 2481, January 1999.

[A6.11] A. Parekh, R. Gallager, “A generalized processor sharing approach to flow control in integrated services networks: The multinode case”. IEEE/ACM Trans. On Networking, 2(2):137-150, April 1994.

[A6.12] J. Bound, et al., “Dynamic Host Configuration Protocol for IPv6 (DHCPv6)”, Internet draft (work in progress), draft-ietf-dhc-dhcpv6-17.txt, March 2001.

[A6.13] H. Schulzrinne, et. al., “RTP: A Transport Protocol for Real-Time Applications”, Internet draft (work in progress), draft-ietf-avt-rtp-new-08.txt, January 2001.

[A6.14] P. Karn., et al., “Advice for Internet Subnetwork Designers”, Intemet draft (work in progress), draft-ietf-pilc-link-design-05.txt, February, 2001.

A7 Simulations Annex

A7.1 Introduction

A7.1.1 Context

This document is the result of the work of BRAIN Workpackage 2 ("Access Network") Activity 2.4 ("Specification of the access network requirements to support the IP-based services").

It is expected that all of the work of this activity will be documented here. The content of this document will eventually become part of BRAIN Deliverable 2.2.

A7.1.2 Scope

The scope of this activity, as given in the WP2 work plan [A7.1], is as follows:

"The main aim of this activity is to support activities 2.2 and 2.3 by simulation of specific aspect as required. This may include the interactions within the access network and external networks."

Work in this activity was divided in 3 tasks:

?? Task 1 – Simulation Framework:
'Background' activity. This includes the implementation of the framework for the simulator. The input of the activity 2.3 will be added to this framework in the later stages.

?? Task 2 – Performance evaluation of protocols:
Includes the simulation of specific protocols and components as required from Activities 2.2 and 2.3.

?? Task 3 – Validation of proposed architecture:
Based on the criteria defined in activity 2.1 the proposed architecture will be validated.

The work in activity A2.4 can be seen schematically in

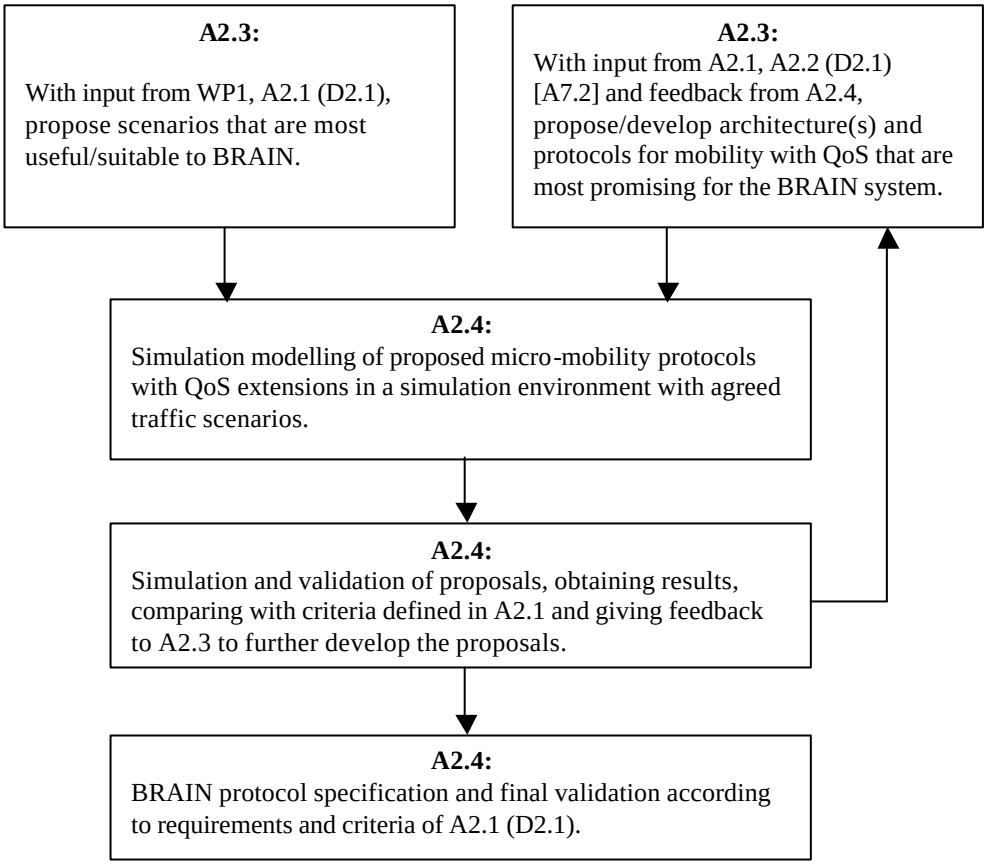


Figure A7-1: Activity 2.4 Workflow

This document therefore describes the work done in simulating various parts of the BRAIN access network as desired from activity A2.3 and the validation of the proposed protocols and mechanisms according to the requirements and criteria set out in activity A2.1.

A7.1.3 Relationship to Other Activities and Workpackages

The technical content of this document relates to other parts of the overall BRAIN project as follows:

?? Activity 2.1: This activity has defined the requirements used to design the various BRAIN protocols and mechanisms and the various criteria against which they are validated through simulations. Furthermore network topologies and traffic scenarios used in simulations were described firstly in this activity in coordination with WP1.

?? Activity 2.2: This document will describe the evaluation through simulations of certain network components (e.g. protocols) for use to support different functions. Thus, it will identify weaknesses and strengths and these results will be used in activity 2.3 to propose a specific protocol or set of protocols that it must be possible to use.

?? Activity 2.3: In effect, this activity determines the work carried out and consequently the input and output (results) of the simulations, as they are described in this document.

?? Workpackage 1 : User services and applications affect the network topologies and the various traffic scenarios and models that were used in the simulations described in this document. As mentioned before activity 2.1 has initially considered these network parameters in the WP2 context. Furthermore the Service Interface between the application and the transport and network layer as defined jointly by WP1 and WP2 will be evaluated.

?? Workpackage 3 : The IP2W interface between the network and lower layers as specified in A2.3 is simulated and evaluated in A2.4 and results are presented in these documents. Furthermore physical layer parameters are used as input to the simulation scenarios..

A7.1.4 Structure

This document is organized as follows:

Section 1.2 gives a brief introduction to the tools that are available when considering simulation of networks. Section 1.3 presents a simulation framework that should be considered for building valid functional simulations. Section 1.4 presents the performed simulations and the relevant results.

A7.2 Simulation tools

A7.2.1 Introduction

Simulation provides a means to avoid costly implementation of communication networks to experiment new protocols. It allows the evaluation of network protocols under varying network conditions in a virtual network environment. Studying protocols, both individually and as they interact with other protocols, under a wide range of conditions is critical to explore and understand the behaviour and characteristics of these protocols. Network simulator (ns) , network animator (nam) visualisation tool [A7.3], [A7.13] and topology generators developed by the VINT project provides several critical innovations that broaden the range of conditions under which existing and proposal of new protocols can be evaluated while making this experimentation tractable. Furthermore, several engineering issues have substantial impact on a simulator's usability. One of these issues is the availability of a wide range of protocol modules in the simulator, which allows easy comparison of different approaches and reduces simulation development time helping the researcher to focus on those aspects of the simulation relevant to their own network design.

The development and use of network simulation has a very long past. Ns itself was a derivative of REAL [A7.4] developed by the Computer Science Department at Cornell University. REAL is also a derivative of NEST [A7.5](a NEtwork Simulator Testbed developed by the Computer Science Department at Columbia University). All the other relevant network simulators cannot be mentioned here, but the distinguish network simulators features will be described in this section in addition to the comparison of some prominent network simulators with ns.

Simulator developers have wide variations of specialisations, where a number of simulators target at a research interest on a particular network type or protocol such as LAN. Whereas simulators like ns, REAL, OPNET [A7.6] and INSANE [A7.7] are aimed at modelling a wider range of protocols. The vast majority of simulators generally provide a simulation language with network protocol libraries: for example, Maisie [A7.8] and OPNET . Highly specialized simulators model or simulate only the details the developer requires.

The core of the majority of simulators, including ns, is a discrete event processor. The accuracy, performance and scaling are important to these simulators, hence several complementary steps are taken to improve upon them. One of these steps, which a few simulators implemented, was to extend the event processor with analytic models of traffic flow or queuing behaviour for better performance or accuracy. Other ways of improving the performance and accuracy are parallel and distributed simulations. Several simulators (such as REAL, Maisie and TeD [A7.9]) support multi- processors or networks of workstations for better performance. Although ns is focused only on sequential simulation, the TeD effort has made some ns modules run in parallel.

Another common approach to improve the simulator's performance, which ns also employ [A7.10], is support for several levels of abstraction. All simulators adopt some level of abstraction given the choice of what to simulate. An example of this is FlowSim [A7.11], which was the first network simulator to make this trade-off clear.

The simulation interface varies from one simulator to another, which includes: programming in a high-level scripting language, programming in a more traditional systems language [A7.8] or both [A7.6] Some simulators focus on permitting the same code to run in simulation and in an actual network, for example, xSim [A7.12] and Maisie. Other simulators emphasise on programming with a GUI shell of some sort, but ns provides a split-level programming model where packet processing is done in a systems language while simulation set-up is prepared in a scripting language. Nam [A7.13] (Network Animator) provides visualization output and is constantly being enhanced to support simple scenario editing.

Concluding, there are numerous options regarding simulation tools that can be used to perform simulations of networks, commercial packages, educational freeware or even specifically, privately developed software. In BRAIN the tools that were used to perform simulations were OPNET, ns-2 and Seawind. These are briefly presented in the following sections.

A7.2.2 ns-2

ns-2 [A7.3] is a free network simulation program that can be downloaded from the web and is compatible with a number of operating systems. The tool has substantial functionality for simulating different network topologies and traffic models. Ns also has an open architecture that allows users to add new functionality. Ns has been developed at the Lawrence Berkeley National Laboratory (LBNL) of the University of California, Berkeley (UCB). The extensibility of ns makes the tool very dynamic; changes

occur frequently enough that a "daily snapshot" is available. This availability of modules makes ns very suitable for the kind of simulations we pretend to perform in Activity 2.4.

ns-2 can be classified as an event-driven network simulator. It is build upon an extensible background engine implemented in C++ that uses OTcl (an object oriented version of Tcl [A7.14]]) as the command and configuration interface. Thus, the entire software hierarchy is written in C++, with OTcl used as a front end. This architecture makes it very simple to extend and to perform our modifications for simulations within the scope of BRAIN.

A7.2.2.1 Modules Features Description

A number of interesting contributed modules could be used. Most of them related to QoS and MM support. There are a number of modules that could be used for traffic generation, but they are likely to require some modifications. The accuracy of the modules is to be identified yet. Next we comment the most interesting modules for supporting BRAIN simulations in ns, basically concerning QoS and MM. There are many others concerning packet for logging, statistics and traffic generation. These are not commented in detail.

DiffServ support [A7.15]]:
This software extends the functionality of the ns network simulator to enable DiffServ networks to be simulated. This patch was generated for the ns-2.1b6 distribution. It includes:

?? DiffServ code points in IP packets.

?? A conditioner which implements drop of EF traffic and remarking AFx1 non-conformance traffic down to AFx2. It has support for several profiles.

?? A primitive WRR scheduler, it has three different queues EF, AF and BE classes.

?? The module includes pre-built DiffServ nodes and means to easily insert conditioners between a link and a node.

There is another DiffServ module. It is distributed from Nortel [A7.16]. It is very complete including:

?? Complete RIO and WRED implementation for queues.

?? Support for EF & AF PHBs.

?? Implementation of srTCM, trTCM and TSWTCM policers.

RSVP support[A7.17]:
This software extends the functionality of the ns network simulation to support RSVP agents. It is a very mature implementation developed by Marc Greis. It includes:

?? Controlled Load with WFQ

?? Soft state with parameterised timeouts

?? FF reservation style

?? API for programming and collecting statistics

?? Link bandwidth reservation for RSVP

It does not include WF and SE reservation, blockade state, INTEGRITY, ADSPEC and policy control. There are a number of differences between RSVP RFC objects and the ones included in this ns module but they won't affect the simulations results. This module is based on Marc Greis' RSVP/ns and there is a patch for ns-2.1b6

Insignia support [A7.18]:
Complete support for INSIGNIA in-band QoS protocol and statistics. The INSIGNIA simulation environment requires the ns-2 simulator and the CMU Monarch extensions (version 1.1.2).

Wireless extensions [A7.19]:
Developed by the CMU Monarch Project. They include a number of capabilities to support wireless and mobility in ns. They provide elements at the physical, link, and routing layers of the simulation environment. Among the most interesting capabilities we can find:

?? Physical: Modelling of signal attenuation, collision, and capture; and Two Ray Ground Reflection radio propagation model.

?? MAC support: IEEE 802.11 DCF MAC and WaveLAN-I CSMA/CA MAC

?? Network interfaces: Lucent WaveLAN DSSS radio.

?? Routing protocols: DSR, DSDV, TORA and AODV.

?? Complete implementation of ARP

?? Concept of mobile nodes with programmable trajectories.

Mobile IP support [A7.20] :
There is a module supporting Mobile IP in ns2 implemented by Charlie Perkins with the following capabilities (capabilities from RFC2002):

?? Agent advertisement

?? Registration

?? IP-in-IP encapsulation

There are no security checks (all registrations are accepted), no de-registrations (any registration pre-empts earlier ones) and no agent solicitation. There is an extension to that implementation developed by University of Southern California [A7.21] with support for overlapping service areas of base stations, intelligent selection of foreign agents and more, with an improved handoff mechanism. The extensions were tested with recent versions of ns version 2.1b6 and version 2.1b7 and might not work with other versions of ns (in particular, they do not compile with the ns release from 18-Jan-2000, although this should be fairly easy to fix).

HAWAII support [A7.22]:
Very complete support for HAWAII micro-mobility protocol with possible extensions for QoS. The implementation in ns is slightly different from HAWAII draft to accommodate its functionality to ns architecture.

Web traffic generator:
Developed by Tom Henderson (UC Berkeley) for the ns-2, simulates a typical web user using HTTP/1.0

Scenario generators:
The ns scenario generator can be used to create different random scenarios for simulation. There are a number of them, being the most common the UCB/LBNL random scenario generator. It consists of a topology generator, an agent generator and a routing generator. The topology generator can generate topology using a standard graph generator (GT-ITM) and converts the topology graph into ns format. The agent generator can be used to define transport protocol agents , type of sources to be used by transport agents, different traffic models for sources and other parameters used by transport agents. The routing generator defines the routing protocols to be used in the simulation. The options include several types of routing.

A7.3 Simulation Framework

As mentioned before, Task 1 of A2.4 includes the definition of a simulation framework based on which all the simulations in A2.4 will take place. There are various issues that are related to a simulation scenario and have to be predefined in the simulation framework. These issues are:

?? Network topology (campus size network, wide area network,…). This affects the size and the number of radio cells.

?? Data traffic models (fixed rate, variable rate, real-time, non real-time, …)

?? Mobility scenarios (mobile node speed, direction, …)

?? Wired and wireless channel physical parameters.

?? Transport protocol used for different applications (TCP, UDP,…)

?? Macro-mobility protocol (Mobile IP, SIP,…) with QoS extensions

?? Micro-mobility protocol with QoS extensions

It is most likely that different cases will have to be modelled for different network topologies and different traffic patterns. Different protocols will also have to be implemented in order to identify weaknesses and further develop various proposals.

A7.3.1 Network Topology

A7.3.1.1 Introduction

The initial description of the Network Topology for BANs is given in D2.1 [A7.2], section 2.1.2 (pages 25-27). The section describes all the entities of the BRAIN network architecture and their relations to other parts of the network. Using D2.1 the most important aspects influencing the network topology design used in A2.4 is extracted. Figure A7-2 below shows the initial high-level topology of the BAN and its relation to other parts of the overall network as presented in D2.1.

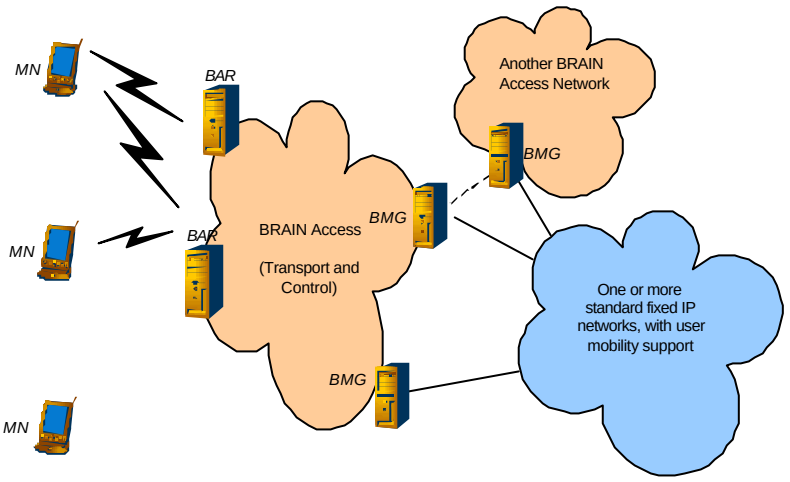


Figure A7-2: BRAIN Network Topology (D2.1)

The key elements of the architecture are listed below with their basic functionalities:

?? **Mobile Node (MN)**: IP host with one or more IP addresses and a single interface and possibly more than one simultaneous radio link with different BARs

?? **BRAIN Access Router (BAR)**: an IP router with multiple wireless and wired interfaces

?? **BRAIN Access Network (BAN)**: data transmission infrastructure and control entities for routing and determining user access

?? **BRIAN Mobility Gateway (BMG)**: special purpose IP router hiding any BRAIN-specific routing functionality

Assumptions and requirements for BANs are specifically important for the design of network topologies. One of these states that a BAN may have multiple interconnections with a fixed network and that it may be connected to more than one fixed network. For very large access networks this translates that, as indicated in Figure A7-2, **BAN should include multiple BMGs for the purpose of avoiding sub-optimal routing to and from MNs and preventing the occurrence of bottleneck, monolithic BMGs**. Therefore, it is essential to include multiple BMGs in some cases of network topologies.

It should be noted that the BAR is the last downlink network layer hop before MNs. BARs use Access Points (APs) (H2 model) to control the wireless link and together form a single IP entity. As far as the BAR is concerned, AP is a "logical" unit with its elements, in particular the AP Controller (APC) (or more correctly Central Controller – CC), which manages one or more AP Transceivers (APTs). However the structure of the AP with parameters such as the number of APTs is transparent to the BAR. The particular structure of the AP determines the wireless link scenario were the RF aspects are controlled by APTs thus incurring the conclusion that the number of APTs maps into the number of wireless cells. Another important issue is the relation of AP to the protocol layer structure inside the BAR. It is considered that the AP "fits" below the IP2W sublayer and this is seen as a network interface. This facilitates the possibility of having more network interfaces inside the BAR where each interface is dealt with a separate AP or another element from a different wireless link technology[87].

A7.3.1.2 Network Scenarios

There are three BRAIN "usage" scenarios defined in D1.1 [A7.23]: nomadic worker, leisure time and medical care using home and school WLAN, public hotspots, on-train public system, corporate system, low range Bluetooth or IrDA and various public networks (GPRS, GSM, UMTS). It was the task of Activity 2.3 to define relevant adequate "network" scenarios to accommodate all "usage" scenarios. These are listed below including physical parameters needed for simulations.

The Small Company

Description	Company internal private WLAN
Scale	small office building with a few workers (< 50 say).
MM	Mobility is low. They are not particularly concerned with vertical handovers to UMTS etc.
QoS / applications	They use it for internal mail and video-mail, www browsing.
Security	used by themselves and perhaps by visitors.

Table A7-1: Small Company Scenario

The University Campus

Description	Campus-wide private WLAN
Scale	A large university campus – many buildings spread over a few km. say 10,000 users D1.1 suggests normal density of 0.25users/m2, but maximum up to 32* this.
MM	Mobility – often none, occasionally medium (walking). Note that many users may change location 'simultaneously' (end of lecture hour) They are not particularly concerned with vertical handovers to UMTS etc.
QoS / applications	Priority of lecturers over students. Email, download of lecture notes, multicasted lectures, on-line games
Security	may not be particularly interested in offering access to visitors.

Table A7-2: University Campus Scenario

The Train Stations

Description	Mix of different access – public WLAN hot spot, offering access to Internet, public WANs (UMTS, GPRS), possibly specialised system on trains, private WLAN (cf small company network scenario)
Scale	say 1km square Say 1000 users
MM	Most users are walking-speed mobility, some are very high speed (on train) Vertical handovers probably important, but note that these are likely to be inter-operator.
QoS / applications	Needs to be billable

[87] Multiple wireless interfaces per BAR would probably result in a scenario where the BAR is the crossover router.

Security	All users are 'visitors' - rapidly changing 'population' (except for private WLAN)

Table A7-3: Train Station Scenario

The Global Network

Description	Global telcos network[88] Many different access technologies Public network + private networks run as a service by the telcos for 3rd party.
Scale	100 million users say
MM	Full range of mobility Vertical handovers required between different access run by operator (e.g. HIPERLAN/2 to UMTS). Inter-operator handovers less critical, though also desirable.
QoS / applications	All applications Needs to be billable
Security	Telco customers + non-customers (cf roaming access)

Table A7-4: Global Network Scenario

A7.3.1.3 Possible Network Topologies

The following topologies represent possible network scenarios in BANs and address several issues:

?? Need for defining the **BRAIN Router (BR)**. This is a wired router with same capabilities as the BAR with the exception that it does not have wireless interfaces.

?? Mapping of network scenarios into network topologies. This should be almost straightforward and obvious although some ambiguity may exist due to the flexibility of number of APTs attached to a BAR as this affects user capacity.

?? APs (and their APC and APT) are assumed to be an integral part of BARs as indicated in Section 1.2. There is one AP per BAR.

?? "Tree versus Mesh" topologies. This was largely influenced by the access networks used for design of micro-mobility schemes. They enforced a "logical" tree topology where regardless of the actual physical topology the routing in the access network is always the shortest path routing to and from Gateways and Base Stations (BMGs and BARs). However, with the introduction of MER-TORA and other MANET-based schemes it is apparent that routing, and associated paths between BARs are an important issue and greatly depend on the physical topologies. Apart from this it was recently observed [A7.25] that performances during handovers are also dependent on the physical topology of the access network. HAWAII's Handover Managements PDI Solution falls short of Cellular IP in some cases of meshed physical topology. Also for the validation of the robustness of some protocol mechanism such as "soft state", it is required to have alternative paths to and from BARs and BMGs such as in the case of mesh topologies.

It is assumed that BMGs are not considered in the Small Company network scenario because of the small size of the network where one BMG is enough. However the set-up of BMGs should be generic and adaptable to any topology even for a small one such as the Small Company case. In an environment where multiple campuses belong to a singe BAN (thus campuses are administratively scoped and advertise an single address space[89]) is it important to deploy multiple BMGs probably as the ingress router for each campus. This is the case with the University Campus network scenario. My thinking at the moment is that for this scenario there should be interconnections between campuses either through BMGs, BRs or BARs thus forming a sole BAN. The campuses can be considered as a collection of

[88] In terms of our BAN architecture, this scenario blurs the BMG / core distinction.

[89] This example requires more research. It can be investigated whether it is sensible to advertise a single address space in a large campus environment (the University Campus) and thus have a micro-mobility mechanism controlling the whole network. This is certainly the fairest solution since it does not discriminate between "bad" and "good" location users (some users can be located at the boundary of campuses and thus experience large delays if micro-mobility is not included in the whole network, although this still depends on the way campus are interconnected and in some cases delays are inevitable due to the distance of "cross-over" router) and my opinion it should be assumed for the time being. Otherwise we need to think of a semi-macro or semi-global handover between campuses.

smaller networks such as the Small Company one with the exception that they do not interconnect with the core network independently but in accordance with the general set-up of the network.

These topologies give example setups for BRAIN simulations. They are as follows:

?? **SM-1 (Small Company-1),** Figure A7-3: This is the basic topology, which incorporates a BAR and a BMG with possible additions of APTs. This topology is probably not very useful as far as the testing of behaviours of BRAIN network layer protocols but presents a model for verifying the flexibility of BAN, which can be adapted to the simplest scenarios such as this one. Additionally, this topology can be a useful simplification of BAN for validation of BRAIN protocols in more complex setups such as the campus environments with multiple BMGs (UC topologies) and for examination of administratively scoped BANs and simplified modelling of macro-mobility or inter-domain handovers.

Figure A7-3: Small Company-1 network topology

?? **SM-2 (Small Company-2),** Figure A7-4: This is a basic "tree" topology. Provides an initial model for testing BRAIN and other network layer protocols. The topology is created in such way that it allows different distances of "cross-over" routers from new BARs (one, two and three hops) in case of standard micro mobility handover.

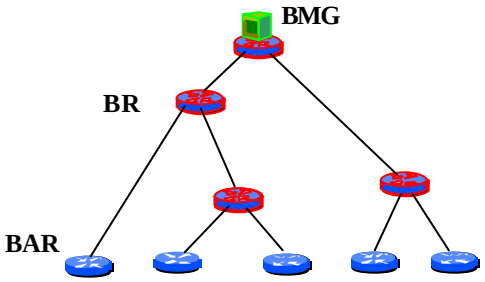


Figure A7-4: Small Company-2 network topology

?? **SC-3 (Small Company-3),**Figure A7-5: Another example of the Small Company topology based on SM-2 but introducing a bus link for examining particular protocol behaviours. Probably irrelevant at this stage.

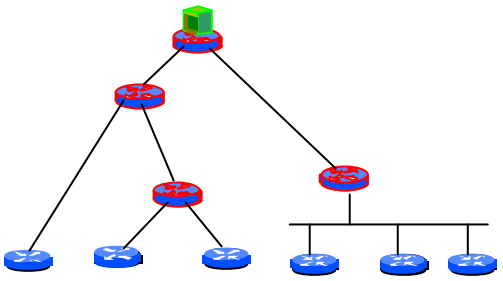


Figure A7-5: Small Company-3 network topology

?? **MESH-1**, Figure A7-6: Due to the extensive range of possible user capacities of this topology (largely depending on the number APTs per BAR) it is not easy to classify this topology in the Small Company, the University Campus or the Train Station case. Therefore it is presented as an independent case because it should be first studied separately in order to check the performance of BRAIN protocols in a sufficiently complex mesh topology. Recommended.

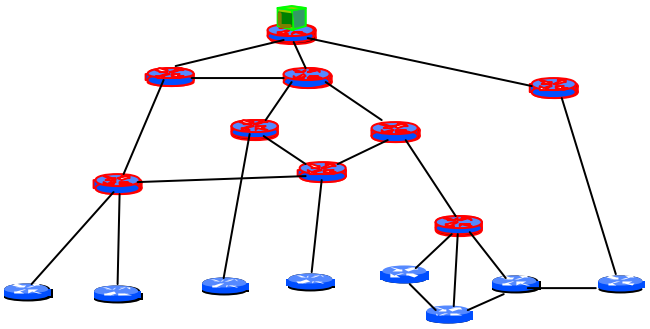


Figure A7-6: Mesh-1 Network Topology

?? **UN-1 (University Campus-1),** Figure A7-7: An example set-up of a small-scale University Campus network scenario which is probably large enough for testing all related behaviours since the maximum capacity case (with up to 10000 users) is not realistic for simulating. This set-up is in accordance with the previously mentioned explanation where a campus is a collection of Small Company and MESH topologies forming a unified BAN. This particular case is a simple one where only BMGs are interconnected. Probably not the best topology for testing coordination of campuses since only BMGs are connected to each other.

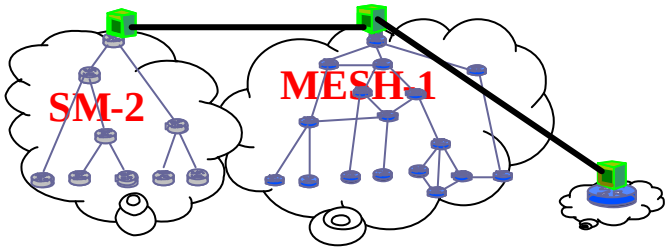


Figure A7-7: University Campus-1 Network Topology

?? **UN-2 (University Campus-2),** Figure A7-8: Similar to UN-1 but an extreme case where BMGs are not directly connected to each other but the interconnections between campuses are achieved though BRs and BARs.

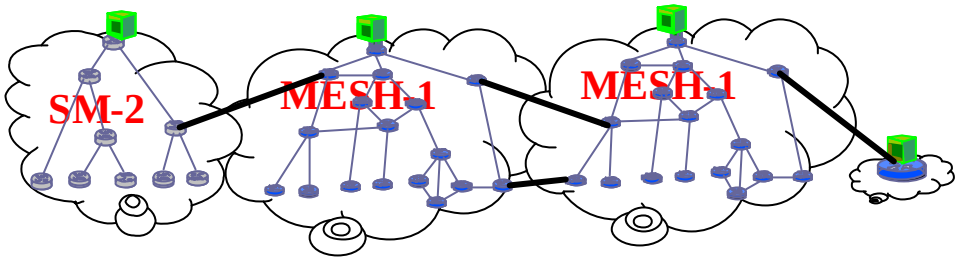


Figure A7-8: University Campus-2 Network Topology

?? **UN-3 (University Campus-3),**: Probably the most realistic scenario with the variety of interconnections of BMGs, BRs and BARs. A recommendation for this case (as it is shown in Figure 10) is to simplify the topologies of the campuses (SMs and MESHs) in order to simplify the simulation and test the overall performance. *Recommended.*

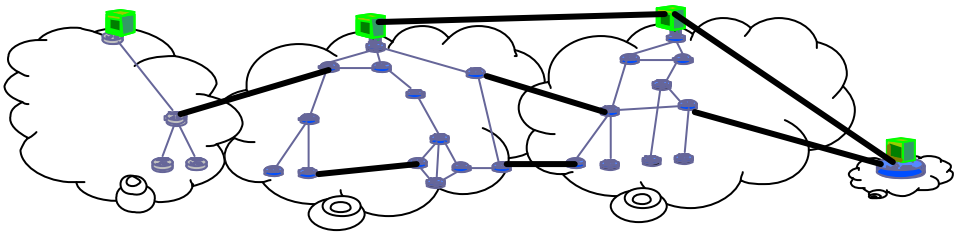


Figure A7-9: University Ccampus-3 Network Topology

?? **TS-1 (Train Station–1),** Figure A7-10: A possible segment of the Train Station network scenario based on the MESH-1 topology. It represents a fragment of the two-dimensional open space shown in Figure 4 (NOKIA). The model used for the University Campus topologies where smaller topologies are included should probably be applies to the Train Station topologies. However, the given topology emphasises that the Train Station has a large number of BARs closely distributed in a single open plane.

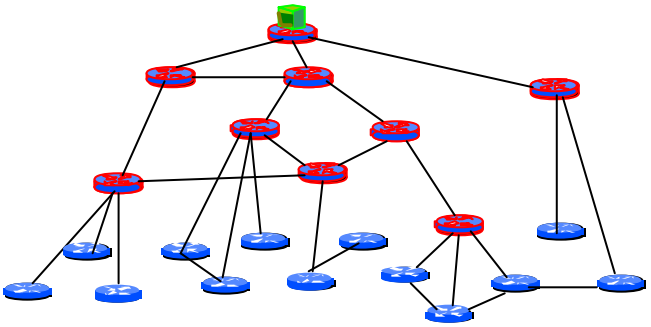


Figure A7-10: Train Station-1 Network Topology

A7.4 Simulation Studies and Results

This section describes the main work carried out in Activity 2.4. It describes the different aspects of the BRAIN access network that where studied and simulated and the relevant results. Studies consider a set of applications generating traffic in a given scenario (section A7.3.1). Different protocol and network topology variables will be used for different simulations. Interaction with physical layer simulations is also considered as part of the IP2W simulation studies.

A7.4.1 BRAIN Candidate Mobility Protocol

This section presents the software implementation of a new candidate micro mobility protocol that was presented in section A3.5 of the deliverable. The protocol functionalities were implemented using ns-2 version 1.b7 and thus the protocol correctness is validated further more some initial results are obtained illustrating the performance of the protocol.

A7.4.1.1 Protocol Simulation

For the purposes of these simulations, Mesh-1 network topology has been used. Following the protocol specification the implemented topology employs one Gateway for the BRAIN Access Network, two Anchor points and four BRAIN Access Routers. The rest of the routers are legacy IP. For simplicity of the code the wireless links are implemented as dynamic wired links, which are turned on and off when a L2 handover occurs, which can then lead to a planned or unplanned handover.

Simulation of the protocol goes through various steps demonstrating its mode of operation:

?? *Phase-phase:* During this phase, signalling messages are exchanged for the configuration of the network with respect to each other.

?? *Login-phase:* A Mobile Host logins through an AR to his closest Anchor, gets an address and informs the CH.

?? *Planned-handover:* Mobile host gets a new AR advertisement and initiates a planned handover through the old AR.

?? *Anchor-update:* During the handover execution phase the Anchor can automatically update the address of the AR that he forwards the data packets to. This is a feature that is controlled by the MH which can manually send an Anchor update at any point.

?? *Unplanned-handover:* When the MH receives an AR advertisement, it requests a handover and switches immediately after it is assigned a new channel.

?? *Anchor-change:* When the MH has reached far from his original Anchor can request a change of Anchor through the anchor closest to the AR he is currently connected to.

An FTP application running on TCP and a CBR application running on two way UDP are established between the CH and MH and are maintained throughout the move of the MH between ARs. The following figures show some snapshots of some key simulation events, as they are animated by nam.

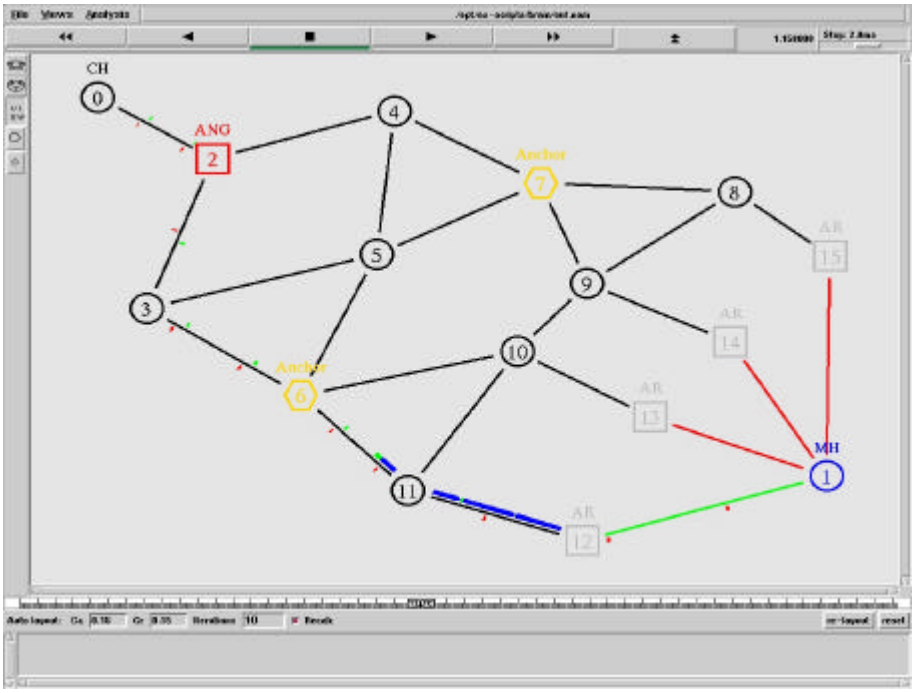


Figure A7-11: Simulator Network topology

Figure A7-11 shows the simulator network topology. The set-up and login phases have been completed and the MH is communicating with the CH. The link between AR in node (12) and the MH is green indicating a live wireless link. The other four links are down indicated by the red colour.

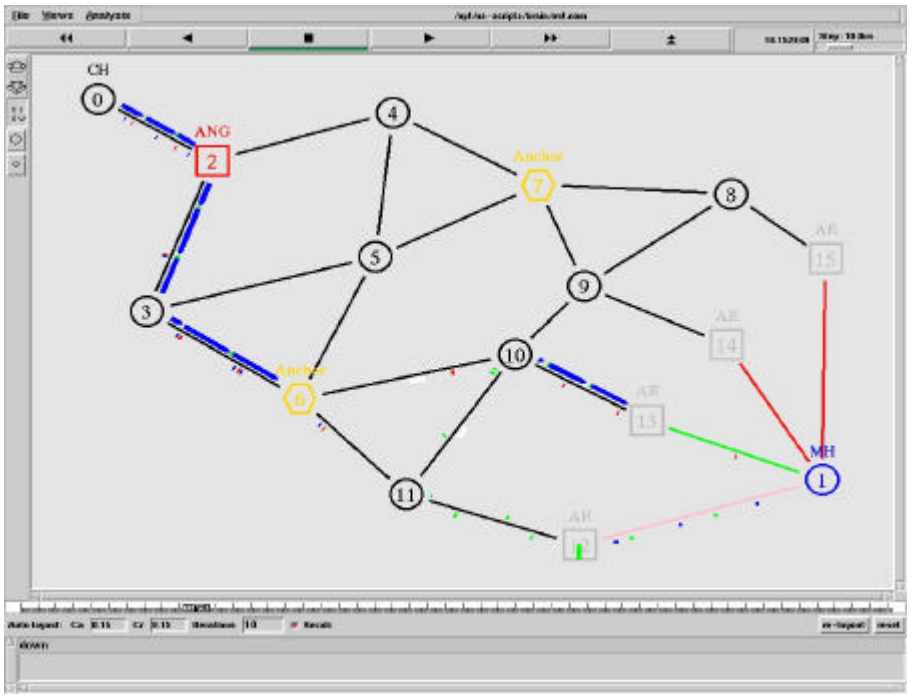


Figure A7-12: Planned Handover

Figure A7-12 demonstrates the planned handover case which is occurring 10 seconds in the simulation. The preparation phase has been completed. MH is communicating with AR in node (13) indicating by the green link receiving packets that are forwarded to him by the old AR in node (12). The previous link is pink indicating that the MH has switched from it, but the old AR is not aware of that and keeps sending the duplicated over the air, which can be seen to be dropped. The white packets are en route handover execution control packets informing the old AR to release its resources and informing the current of the new AR, through which the MH is currently connected, so that he can optimise the routing ..

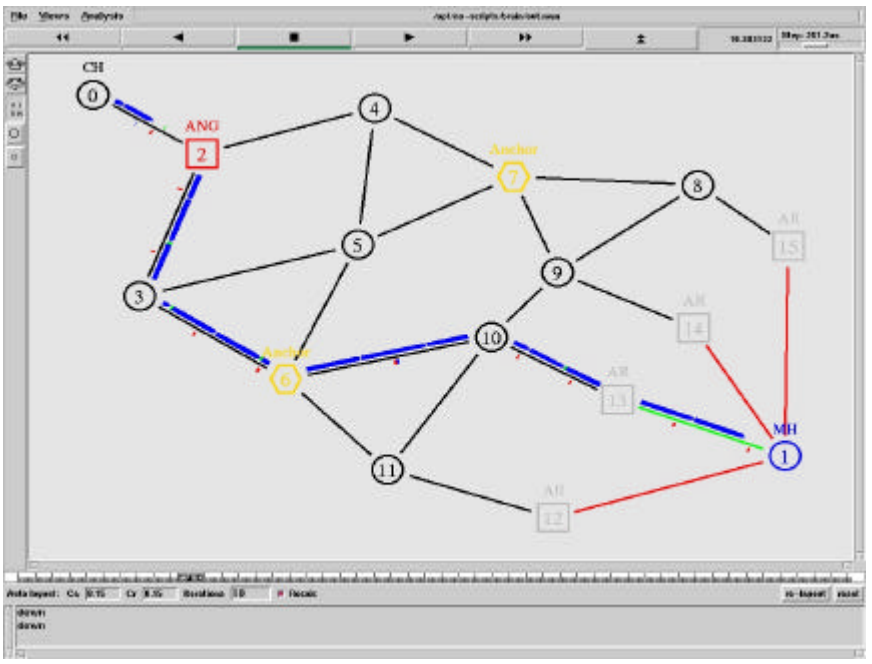


Figure A7-13: Planned Handover Completed and Anchor Optimised.

Figure A7-13 shows the completion of the handover, where the old AR has released the assigned resources and the route to the new AR has been optimised.

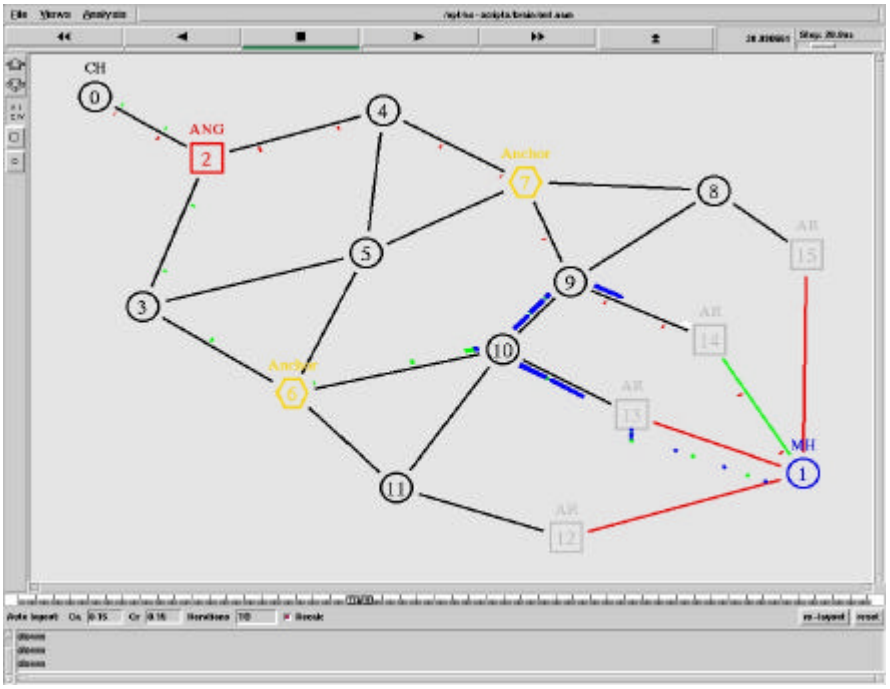


Figure A7-14: Unplanned Handover

Figure A7-14 shows the end of the unplanned handover phase, which occurs 20 secs in the simulation. The link between MH and AR has switched from AR in node (13)(link red) to AR in node (14) (link green). AR (3) has stopped sending over the wireless link since these packets were dropped and Anchor has also been optimised again. It can also be seen that uplinks packets are send with standard routing, finding the fastest way out of the network.

Figure A7-15shows the packet routes after the completion of a second planned handover which occurs 30 seconds in the simulation. Anchor has also been optimised with the new AR address in node (15). Its worth while noticing the fact that packets are send upwards a lot faster that in the downlink, which affects the performance of the data transfer.

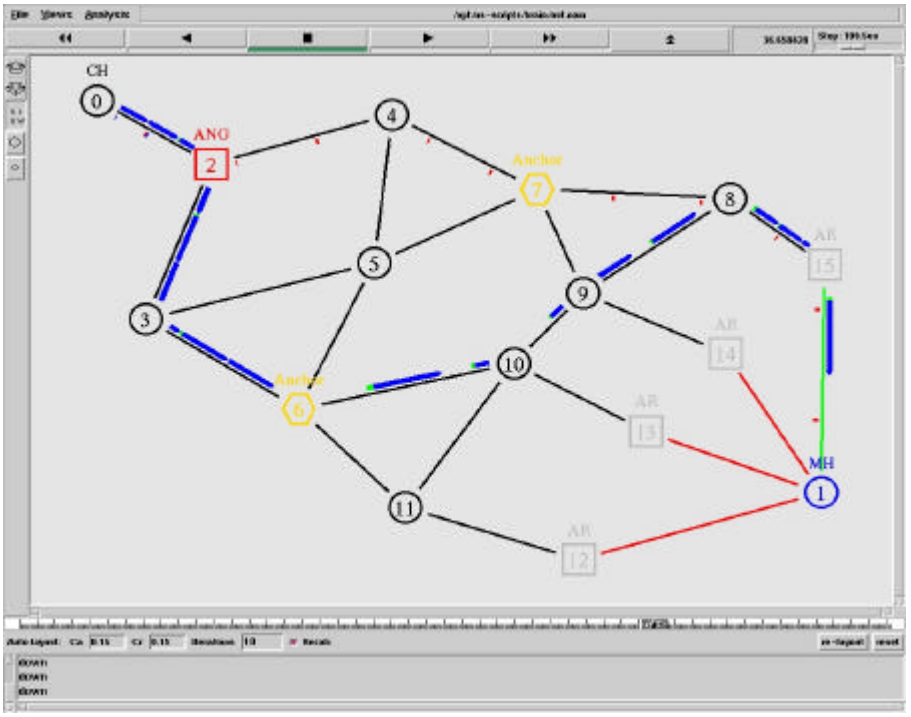


Figure A7-15: After second Unplanned Handover

MH now from the received data packets or hinted by the network realises that it has moved away from his original Anchor and request a change of anchor through AR's, in node(15), closest default Anchor in node(7), 40 minutes in the. Once the registration with the new Anchor has been completed packets are now routed from the new Anchor, Figure A7-16.

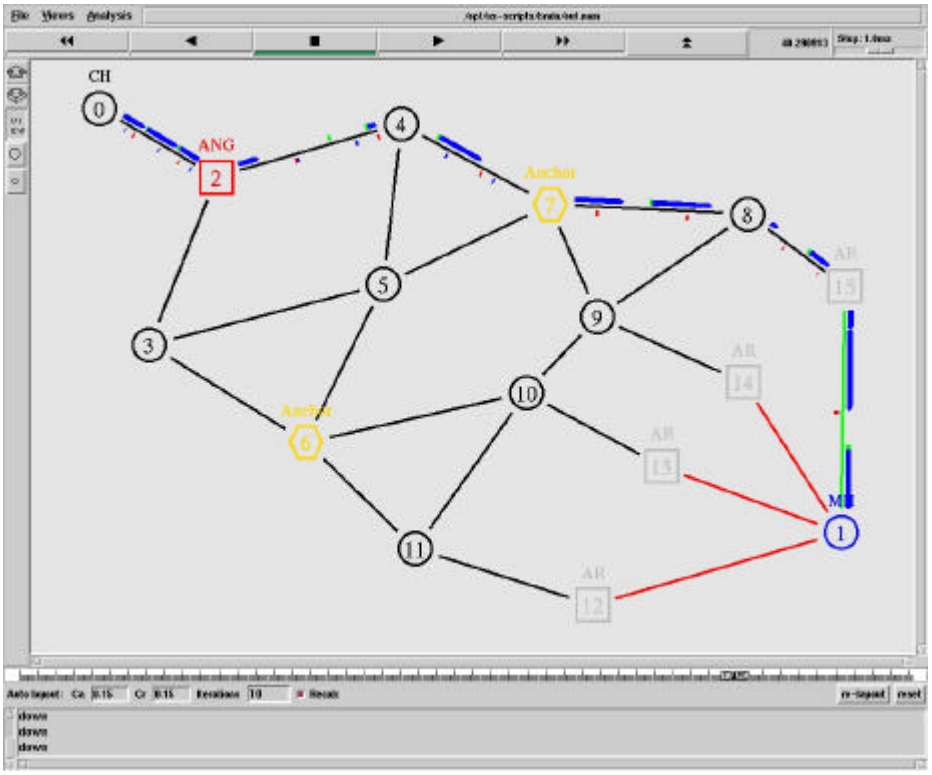


Figure A7-16: Change of Anchor

A7.4.1.2 Protocol Performance

In this section various performance related graphs will be presented showing the effect of mobility on the applications running and the relevant transport protocols. Because the graphs have the simulation time in there x-axis, bear in mind the time that various events are happening during the simulations:
- 10 seconds, planned handover with automatic Anchor optimisation.
- 20 seconds, unplanned handover with automatic Anchor optimisation.
- 30 seconds, planned handover with automatic Anchor optimisation.
- 40 seconds, manual change of Anchor.
- 50 seconds, end of simulation.

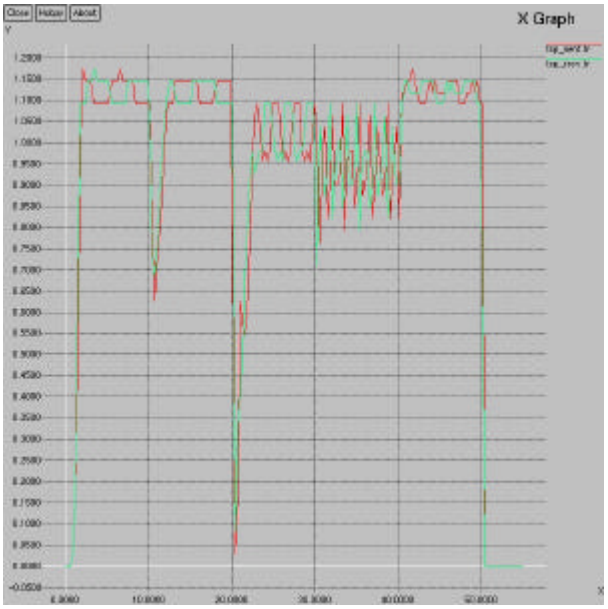


Figure A7-17: TCP Throughput Sent and Received vs. Time

Figure A7-17 shows the TCP throughput for both sent and received packets. The variation of the throughput due to the handovers can be seen. Especially in the case of unplanned handover throughput falls almost to zero before picking up again. The throughput decreases as the MH moves far away from the Anchor. Note that after the third handover throughput varies a lot caused by the different routes that ACKs follow towards the MH. When the MH changes Anchor the throughput returns to the maximum value. Note that this case is not realistic since during a change of Anchor a change of MH address is also involved which means that the TCP connection will be killed and restarted. I any case after restarting it will rapidly rise to the maximum value.

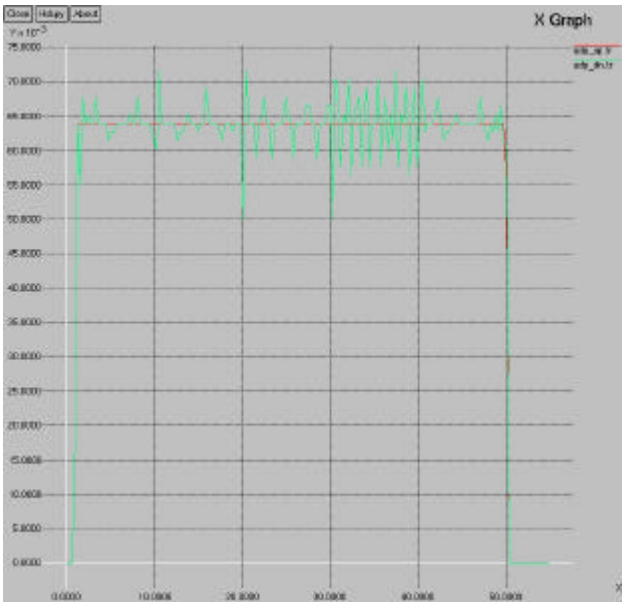


Figure A7-18: UDP Uplink & Downlink Throughput vs. Time

Figure A7-18 demonstrates the variation of average UDP throughput for uplink and downlink. Although uplink throughput is not affected by the handovers, downlink throughput varies a lot during and after a handover, something that is not desirable for some CBR application and that will require some buffering and traffic shaping at the AR.

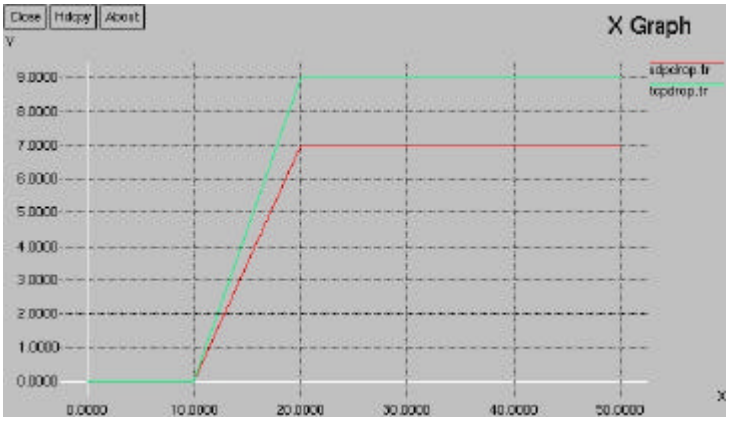


Figure A7-19: TCP (1024 pkt size) and UDP (CBR@64kbps & 48bytes pkt size) Packet Dropping vs. Time

Figure A7-19 shows the number of packets that were dropped during the handovers. There are no packets dropped during the planned handovers and there were 7 and 9 packets dropped for UDP and TCP respectively during an unplanned handover. Note that this graph does not include the number of duplicated packets that were send by an old AR while the MH has switched to the new one and received the redirected ones.

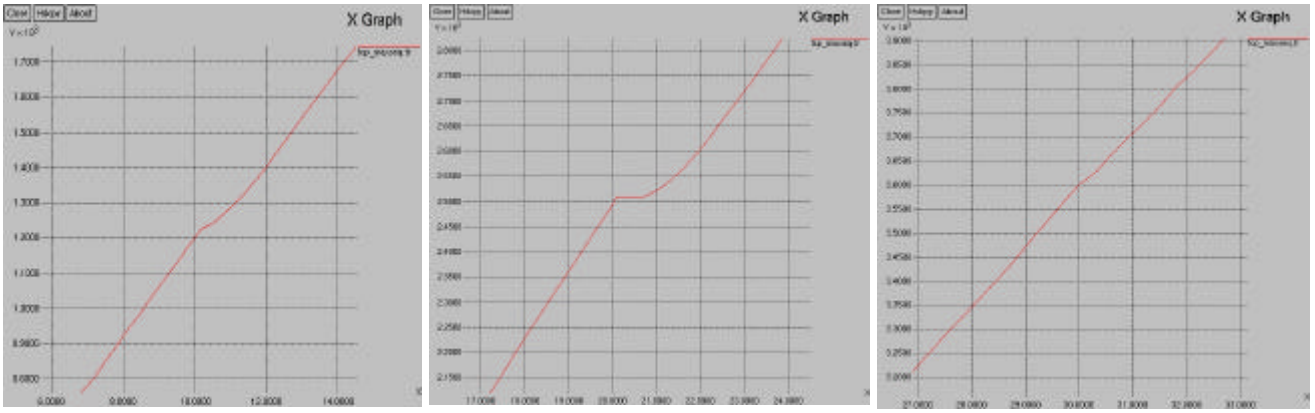


Figure A7-20: TCP Transmitted Packet Sequence Number vs. Time

The above figure show the sequence number of the TCP packet that is transmitted before during and after a handover. The gradient of the graphs shows how fast TCP is recovering from a disruption. The worse case is for the unplanned handover, which then requires almost one second to recover.

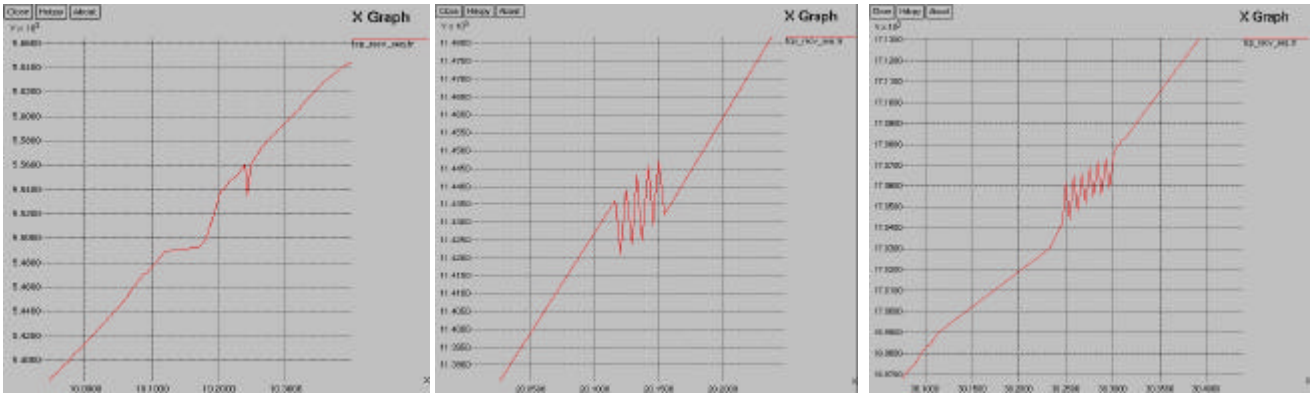


Figure A7-21: Received TCP Packets Sequence Number vs. Time

Figure A7-21 shows the sequence number of the received TCP packets when a handover occurs. Packets irrespective of the type of handover (planned or unplanned) are received reordered but TCP does not respond to that. Note also that even when a planned handover happens when the MH is far away from his

Anchor, the problem is quite noticeable and as the graph show its worse than in an unplanned handover. There are no duplicate packets received, because in this implementation the same handoff-preparation-acknowledgement message, that triggers the current AR to duplicate packets and redirect them to new AR, triggers also the MH to switch L2 connections to new AR, and thus is not listening at the old channel any more.

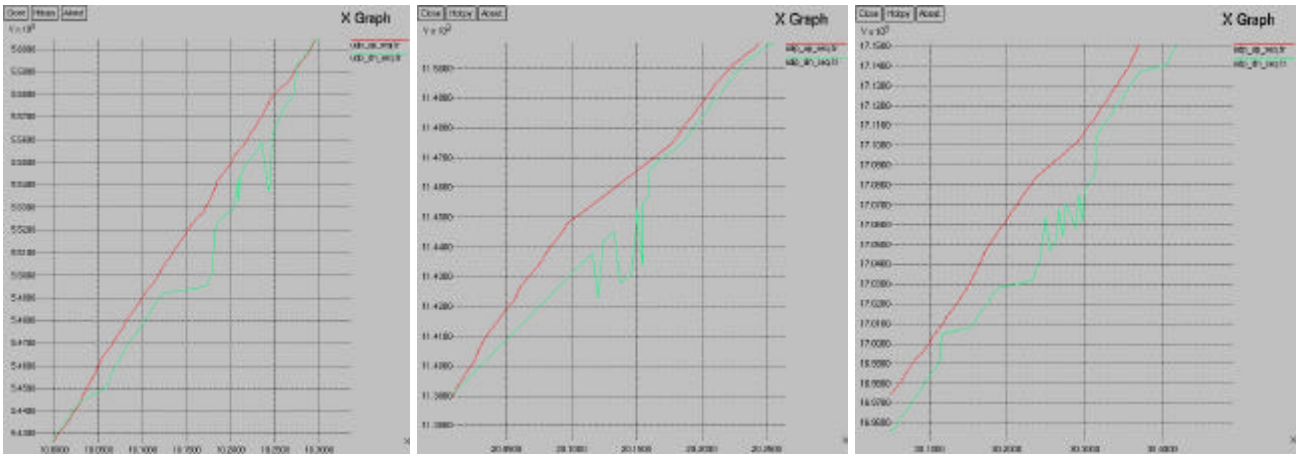


Figure A7-22: Uplink and Downlink Received UDP Packet Sequence Number vs. Time

Figure A7-22 shows the sequence number of the received UDP packets both at the MH and the CH. It can be seen that uplink packets do not get reordered, unlike downlink packets that do especially when the MH has moved far away from his Anchor. This is a similar situation encountered in the case of TCP packets. As it was mentioned before buffering and traffic shaping in AR is required

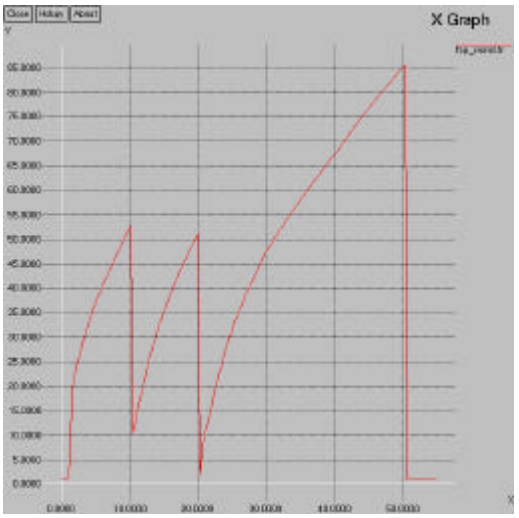


Figure A7-23: Tahoe TCP Congestion Window vs. Time.

Finally Figure A7-23 presents the size of the congestion window and how it is affected by the occurrence of a handover. This graph is not related to the actual mobility protocol rather than to the performance of TCP in these situations. In the first planned handover, TCP triggers the congestion window reduction by a factor of five, even if there are no dropped packets. This suggests that maybe an ACK timeout has occurred. During the unplanned handover the congestion window is almost reset. The last planned handover is undetected by TCP which continues to increase the size of the congestion window. Different TCP schemes could be simulated, in order to decide on the more suitable scheme for an environment like the one considered.

A7.4.1.3 Conclusions

Summarizing, during this simulation task the new mobility protocol that is proposed as a candidate mobility protocol, was implemented using ns-2. Most of the protocol features, except paging (to be included) and log out (trivial), were included in this implementation and the operation of the protocol and its signalling was validated. Animation screenshots were provided showing an example network topology that was used. Protocol performance was then evaluated for different applications and transport protocols

and demonstrated using graphs showing throughput, packet dropping, packet reordering, packet retransmission, etc… Planned handovers are smooth, as they are fast and no packet dropping occurs, whereas during unplanned handovers packets are dropped but system recovers fast. Another problem was identified, when the MH has moved away from the original Anchor (with or without a handover occurrence), since uplink packets reach the source faster than downlink packets, which result in a throughput variation. This suggest the need for fine tuning of the change of Anchor functionality of the MH. Finally there is also a need for solution to the problem of reordered packets reaching the MH which can influence some (CBR) applications. In general it is concluded that this is a sound, functional proposal with good performance. Future work includes the simulation of larger and more realistic network topologies with more traffic, which will enable the positioning of the Anchors in different levels and testing the performance of the protocol.

A7.4.2 Hawaii simulations

This section presents the simulations performed with Hawaii protocol in the same scenario that was used previously with BCMP. The topology and the traffic patterns are the same, thus allowing the comparison of results. Firstly the behaviour of Hawaii in our topology will be described using some screen snapshots and then its performance will be shown. The performance results will be analysed against the BCMP performance.

A7.4.2.1 Hawaii behaviour

The behaviour of Hawaii in the scenario described in section A7.4.1 will be shown using some screen snapshots. Each one corresponds to key simulation events, animated by nam, and explained after each picture. It is important to remark that we always used the Unicast Non Forwarding mode of Hawaii.

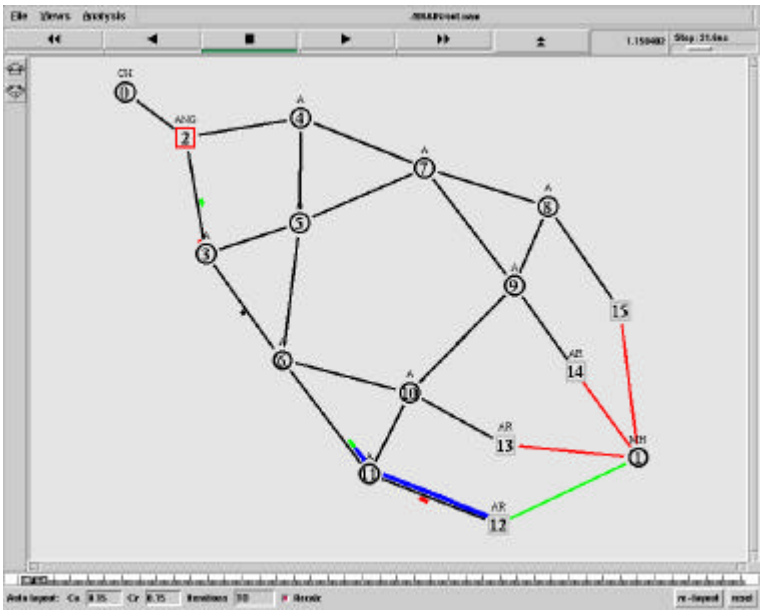


Figure A7-24: Simulator Network Topology

Figure A7-24 shows the simulator topology. The mobile host (MH) labelled as node 1 in the figure is communicating with the correspondent node (CN) labelled as node 0. The up link and down link have been established using the shortest path through nodes 2-3-6-11-12. Note that the link between node 12 and 1 is green indicating an active wireless link, while the other wireless links are coloured red indicating that they are down. Traffic packets are coloured depending on the flow they belong to; TCP packets and their ACKs are coloured in blue, up link UDP packets are coloured in red while down link UDP packets are coloured in green. Hawaii packets are coloured in black.

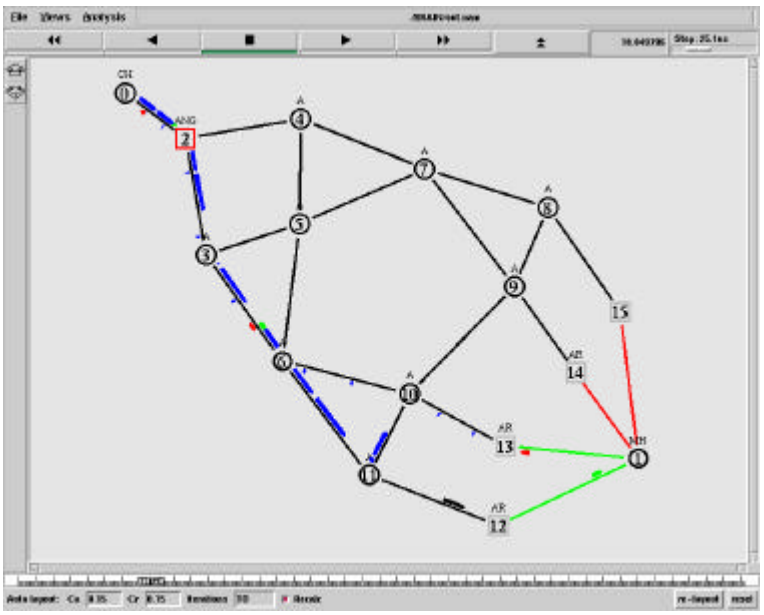


Figure A7-25: Planned Handover

Figure A7-25 illustrates the *planned* handover. Hawaii is not supporting the planned and unplanned handover as BCMP does. Thus here, planned handover means that the MH is connected during the handover to the new and old base stations at the same time. This implies that no packets will be lost during handover. Later we will shown the unplanned handover, where the MH changes abruptly from the old base station to the new one. In this case, there will be some packets dropped; this is the way the UNF mode of Hawaii works.

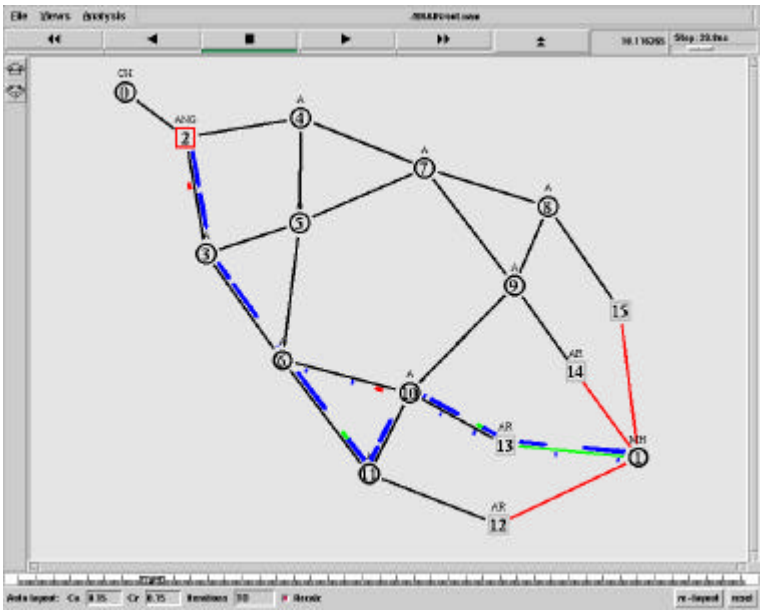


Figure A7-26: Completed Planned Handover

In Figure A7-25 the planned handover is almost completed. The MH is already attached to the new access router (node 13) and sending the up link traffic through it. The handoff message (black packets in the link 11 to 12) has already reached the crossover router (node 11) producing the route update. Thus the down link traffic (blue packets) are already forwarded to the new access router. At the same time, the MH is still receiving packets through the previous access router (node 12).

Figure A7-26 shows the planned handover completed. Resources are already released at the old access router station (node 12) and the down link traffic is forwarded to the new access router at the crossover router (node 11). Note that the up link traffic, both UDP (red packets) and TCP acknowledges (small blue packets), follow the default route, the fastest way from the MH to the CN.

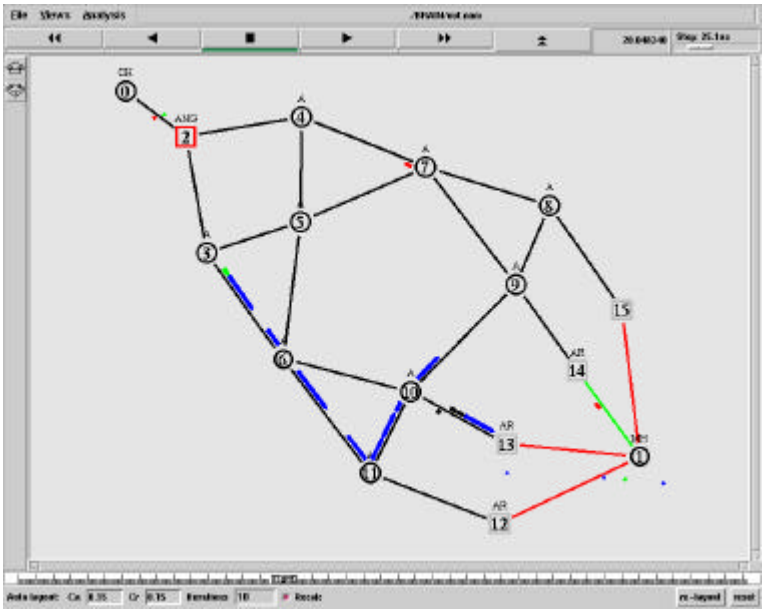


Figure A7-27: Unplanned Handover

Figure A7-27 illustrates the final phase of the unplanned handover. The MH has abruptly changed its point of attachment, producing the dropping of packets in the wireless link. Additional packets travelling from the crossover router to the old access router will also be dropped when arriving at the old access router. Note also that the new access router sent the handoff update message to the old access router (shown in the picture as black packets arriving at node 13). This message produced a route update in the crossover router (node 10), thus packets are now forwarded to the new access router (node 14).

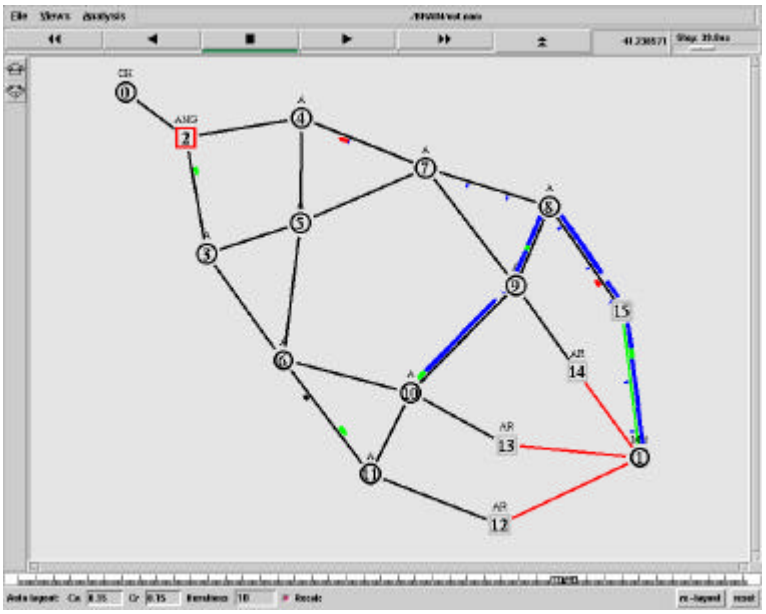


Figure A7-28: After Second Unplanned Handover

Figure A7-28 illustrates the final situation, after the second planned handover to access router at node 15. Up link traffic follows a shorter path than down link traffic. This is a direct consequence of the fact that Hawaii does not incorporate any route optimisation. When the mobile host moves away from its first point of attachment, the down link traffic follows a sub-optimal route.

A7.4.2.2 Hawaii performance

In this section performance graphics equivalent to those at section A7.4.1.2 will be presented, allowing the performance comparison between Hawaii and BCMP. It is important to remember the time when main events happened, as the time is shown in the x axis. There was planned handovers at 10 and 30 seconds, and one unplanned handover at 20 seconds. There is no anchor optimisation at 40 seconds, because Hawaii does not incorporate that functionality.

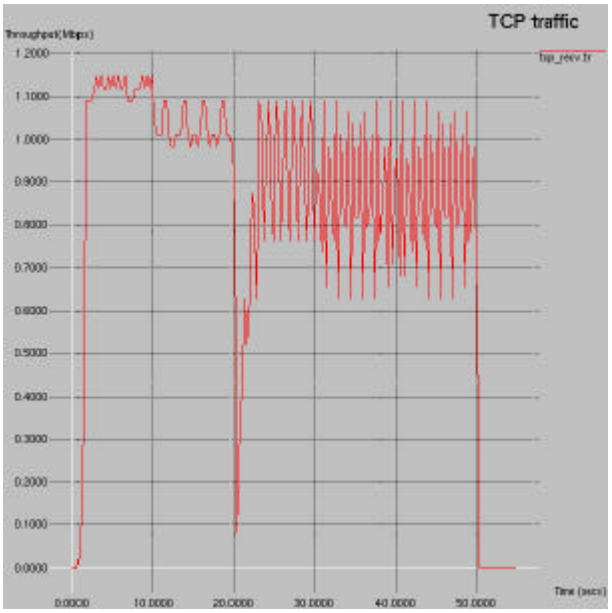


Figure A7-29: TCP Throughput Received Traffic vs. Time

Figure A7-29 shows the TCP throughput. Before the first planned handover at 10 seconds, the throughput is the same than in the BCMP case. After the first handover, the throughput follows the same pattern than in the BCMP case but its value is slightly lower here. This is due to the longer path of the downlink TCP packets as the MH moves away from its initial point of attachment. This is specially severe after the BCMP anchor change at 40 seconds. At that time, the BCMP recovers the full speed of the TCP flow, while here we can see that no TCP speed increase is perceived after the second planned handover at 30 seconds. TCP throughput in the Hawaii case is lower compared to the BCMP case because the longer total round-trip time limits the speed at which congestion window is increased. If the simulation had run long enough, both throughputs would have been the same. Additionally the maximum bandwidth is not used because the congestion window limits the TCP throughput.

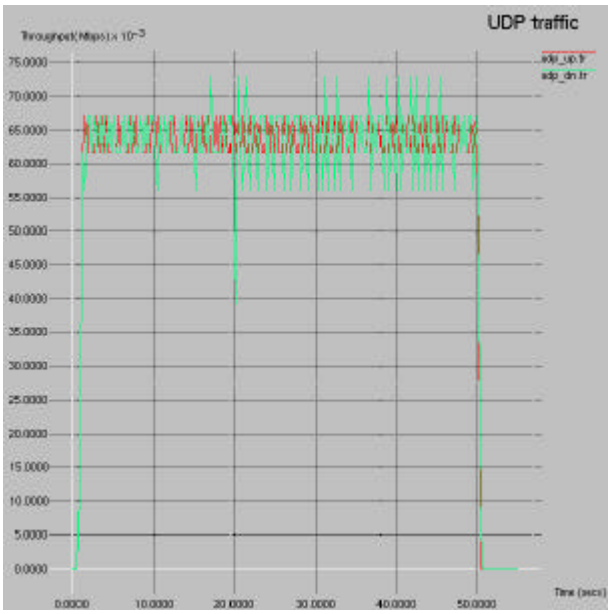


Figure A7-30: UDP Uplink and Downlink Throughput vs. Time

Figure A7-30 shows the UDP throughput in both directions. The performance here is similar to the performance of BCMP. The up link traffic is not affected by the handovers, while the down link traffic is affected only during the unplanned handover at 20 seconds. The down link traffic also suffers a slightly longer transmission delay than in the BCMP case because the longer path it has to travel.

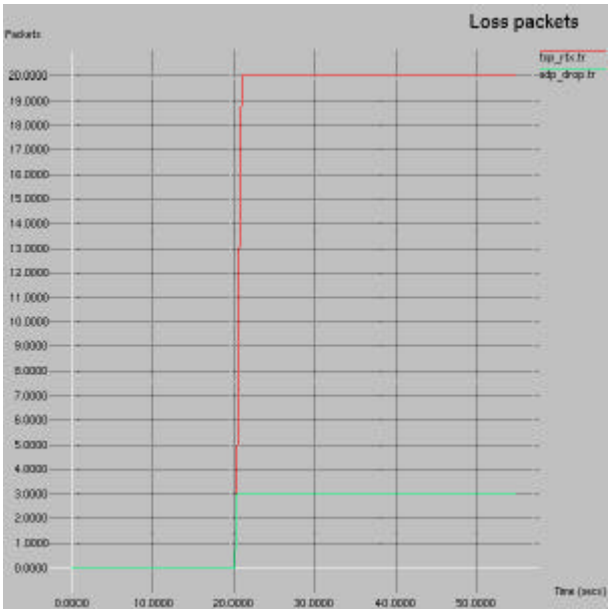


Figure A7-31: TCP and UDP Packet loss vs. Time

Figure A7-31 shows the loss packet for both TCP and UDP traffic. Loss packets include dropped packets and packets arriving too late in the TCP case (TCP timeout). It is important to highlight that no packet loss is detected during planned handovers, while several packets are dropped (or arrived too late) in the unplanned handover. This loss packet rate is higher than the BCMP one. This explains why the TCP connection takes longer to recover its full speed with Hawaii.

Figure A7-32: TCP Maximum Transmitted Sequence Number

Figure A7-32 presents the TCP maximum transmitted sequence number before, during and after each of the handovers. Recall that the gradient shows how fast the TCP is recovering after a disruption. During the planned handovers, first and third figures, the sequence number is not affected. Nevertheless, the TCP is rising its throughput slower in the third figure due to the longer rtt (increased down link path). The second figure shows how the sequence number is affected by the unplanned handover. It takes almost 1 second to recover after the handoff. In general, the gradient of this graph is lower than the BCMP one, indicating that the TCP connection recovers its throughput better using the BCMP.

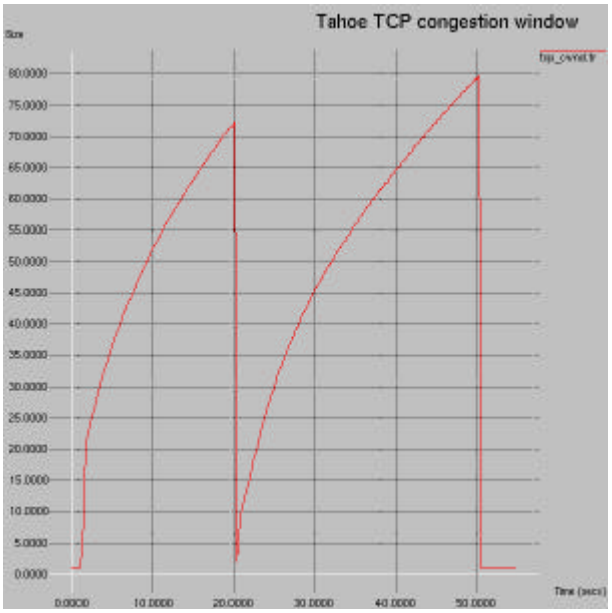


Figure A7-33: Tahoe TCP Congestion Window Size vs. Time

Figure A7-33 presents the TCP congestion window size and how it is affected by the handovers. Planned handovers happen unnoticed to the TCP connection, while unplanned handover produces not a reduction but a reset of the connection window. The increase pattern of the congestion window is similar to the BCMP one, but it reaches a lower upper value here.

A7.4.3 Hawaii and RSVP Coupling Simulations.

This section presents the simulations performed with Hawaii and RSVP protocol in the Small Company 2 scenario (SC-2). The scenario is explained in detail in section A7.3.1.

First the behaviour of Hawaii and RSVP in this topology will be showed with some screen snapshots. After that the performance improvement of the loose coupling of RSVP and Hawaii will be showed. The procedure includes the comparison of some performance parameters such as delay, loss and throughput when both protocols are coupled and de-coupled.

A7.4.3.1 Scenario

Among the possible BAN network scenarios, we have selected the small company network topology (SC-2) in our simulations.

SC-2 is a basic “tree” topology. It provides an initial model for testing BRAIN and other network layer protocols. The topology is created in such way that it allows different distances of “cross-over” routers from new BARs (one, two and three hops) in case of standard micro mobility hand-over.

Inside the BAN we have considered duplex links with the following parameters values:

?? Bandwidth: 512 Kb

?? Delay: 10 ms

Notice that delay depends strongly on the network technology so this value may vary.

HIPERLAN/2 links are modelled using Nokia link layer simulations. We will approximate each cell using the industrial hall with no internal walls contributed by Nokia. Nokia has evaluated the HIPERLAN/2 air interface behaviour for different load levels. The hall size is 250 metres by 250 metres and we have a load level similar to 5 new FTP transmissions per second.

We have characterised the HIPERLAN/2 link as two fixed simplex links (up and down) with three parameters to be determined: delay, bandwidth and loss model. For the selected load level, link parameters are approximated as follows:

?? Bandwidth: 3.2 Mb

?? Delay: 15ms

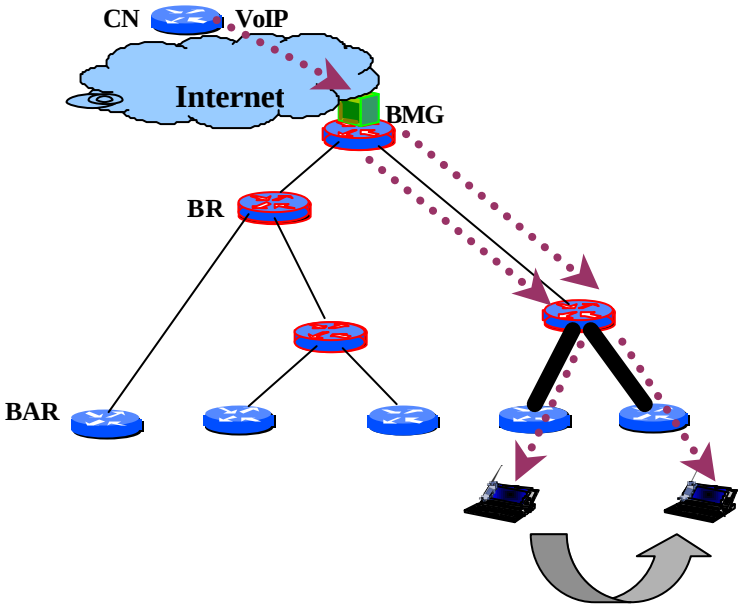


Figure A7-34: Network Scenario

We have located the correspondent node outside the BAN. It is just one hop away from the BMG although it could be located in any place in the Internet. It is sending VoIP traffic towards a mobile node inside the domain. We will consider that the mobile node changes its position between consecutive cells during the call time.

The simulation is performed as follows: during the first 100 seconds the correspondent node performs the reservation and begins transmitting voice packets towards the mobile node. Afterwards handover between consecutive cells takes place. This implies a modification of the routing tables using HAWAII and the necessity to reserve bandwidth across the new path with RSVP messages. As we will se, if we want to optimise network resources then we have to couple both protocols in order to not waste extra time making the reservation after the handover occurs.

Links between the intermediate node and the base stations are loaded up to 100% by background traffic in order to show the benefits of reservation with RSVP for the voice traffic. These links appear in black in the figure above. Background traffic is characterised as constant bit rate traffic.

The speech traffic model extracted from D3.1 can be described as a birth-death process with a Poisson distributed arrival process and a exponential distributed call duration. In a conversation each party is alternating active and silent. During the activity phase IP packets carrying the speech information are transmitted. We are going to simulate this traffic considering activity and silent periods are generated by an exponential distributed random variable. The mean value of this variable will equal Ton during activity periods and Toff during silent ones.

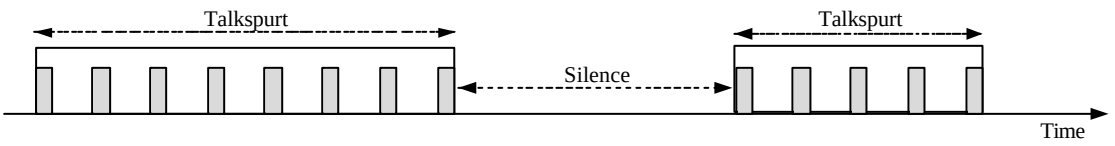


Figure A7-35: VoIP Traffic

VoIP traffic model main parameters are shown below:

?? Activity interval: 50 %

?? Mean call duration: 120 sec

?? Mean active phase T_on: 3 sec

?? Mean silent phase T_off: 3sec

?? Payload of IP packet: 32 Bytes

?? IP packet rate: 12.2 KBPS

In telephony applications the transport delay should preferably be less than 100 ms.

In order to generate traffic with similar characteristics to VoIP we use a special traffic generator implemented in ns-2. The constant bit rate generator generates traffic according to a deterministic rate. In the real case packets are sent at a fixed rate (12.2-Kb) only during on periods, and no packets are sent during off periods. We are going to assume on period take 100% of time so we can use a 12.2 Kb CBR source transmitting all the time. Packets are constant size.

This application sits on top of a transport agent, in our case UDP. We can change our traffic source characteristics by varying its two parameters: packet size and rate. We have implemented a simplified model. The mobile node keeps in silence while the correspondent node transmits continuously. We measure the delay associated to the VoIP packets that travel one way. This assumption is correct since links have different queues for the different ways. Packets from the CN don't interfere with packets coming from the MN, so the delay obtained for that link sense is correct.

The simulations have been performed using ns2 release b5. We have installed two patches: RSVP and HAWAII, and we have observed the interaction between them. In order to make them work together correctly we have modified some aspects of the code.

A7.4.3.2 Behaviour of HAWAII and RSVP in this scenario

The implementation of HAWAII used is identical to the showed in section A7.4.2, where HAWAII is evaluated against BCMP. The only difference is that the scenario is simpler and only one handover is performed. Situation before and after the handover can be seen on the following pictures.

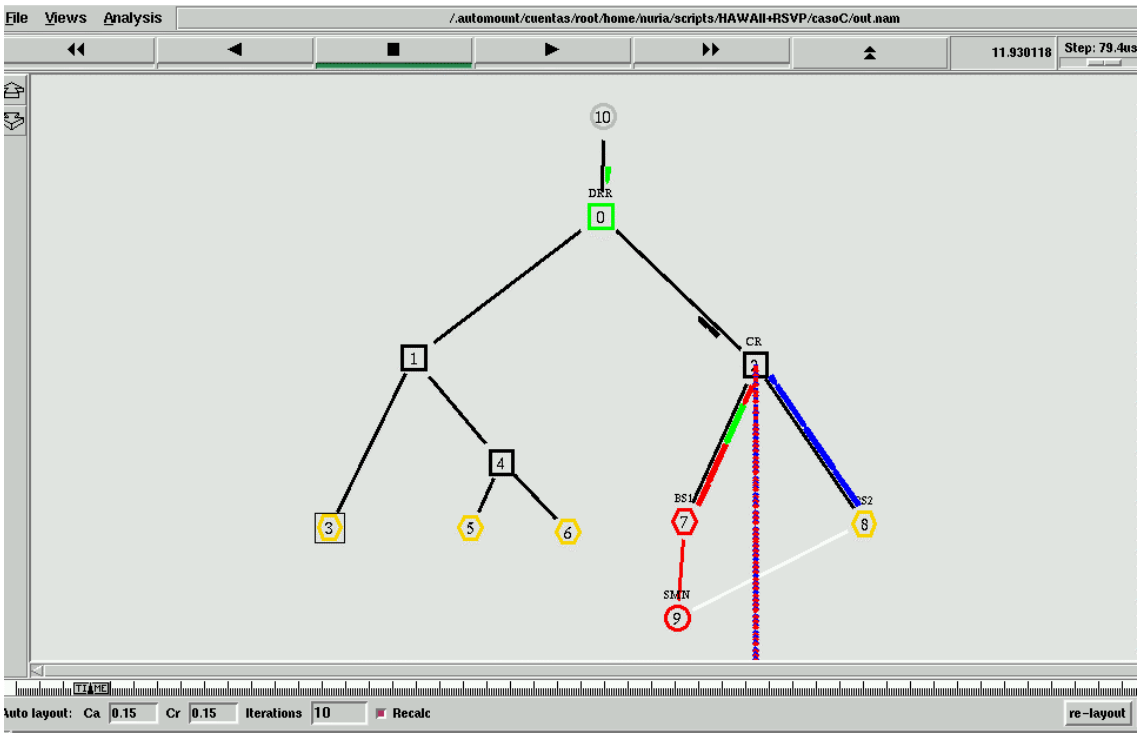


Figure A7-36: Simulation Network Topology

Figure A7-36 shows the simulator topology. The mobile node (MN) labelled as node 9 in the figure is communicating with the correspondent node (CN) labelled as node 0. The two base stations are labelled 7 (BS1) and 8 (BS2) respectively. The up link and down link have been established using the shortest path through nodes 10-0-2-7-9. Note that the link between node 7 and node 9 is red indicating an active wireless link, while the other wireless link is coloured white indicating that it is down. Interference traffic packets are coloured red and blue, they go from node 2 to 8 (blue) and from node 2 to 7 (red) just to interfere with voice traffic and show the behaviour of RSVP reservation. Hawaii packets are black and RSVP packets are yellow. In the figure, note that a lot of interference traffic is being discarded because the links don't have enough capacity. Voice traffic is green and it is not being discarded because the

reserve is already established. Note that in these snapshots voice traffic rate is considerably low (12 kbps) so only one or two VoIP packets are shown in one snapshot.

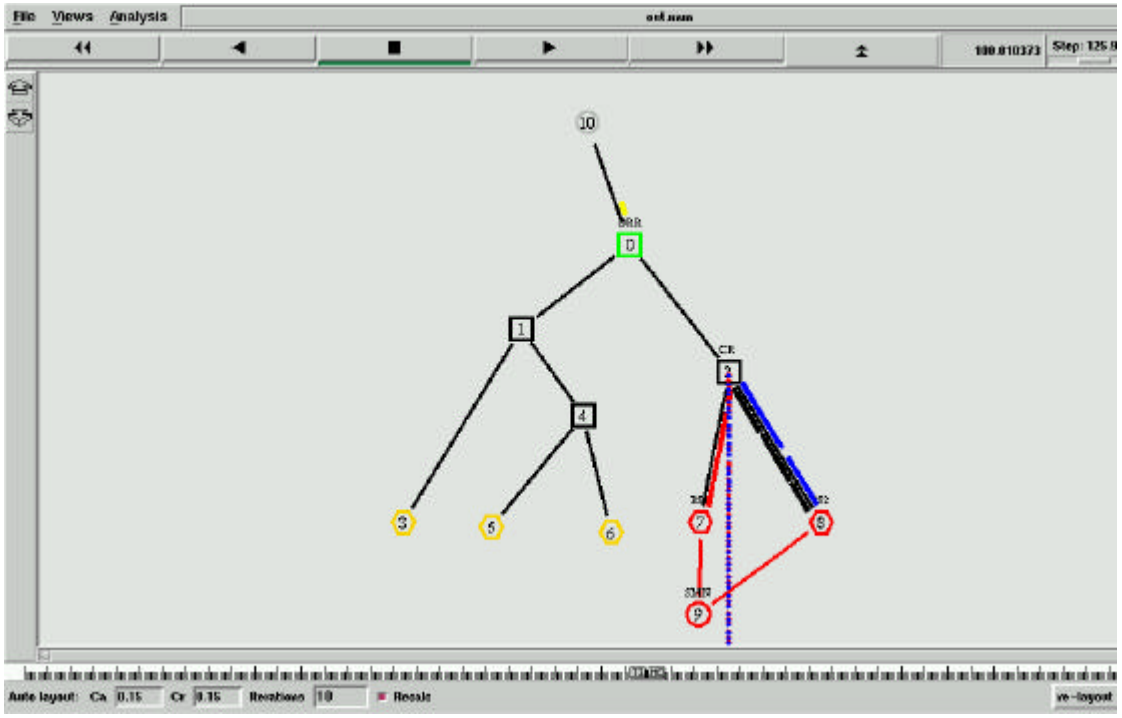


Figure A7-37: Handover

Figure A7-37 shows a planned handover. In this case study we will only see the effect of the coupling of protocols during a planned handover. The MH changes its point of attachment, and for 20 ms both wireless links are active (in red) while Hawaii packets (showed in black) sent by the new access router carry the handoff update message to the old access router. This message produced a route update in the crossover router CR (node 2), thus packets are now forwarded to the new access router (node 8). Note that although, in fact it is a planned handover, the amount of time during which both links are active (20 ms) is rather short, and that has a major impact on performance.

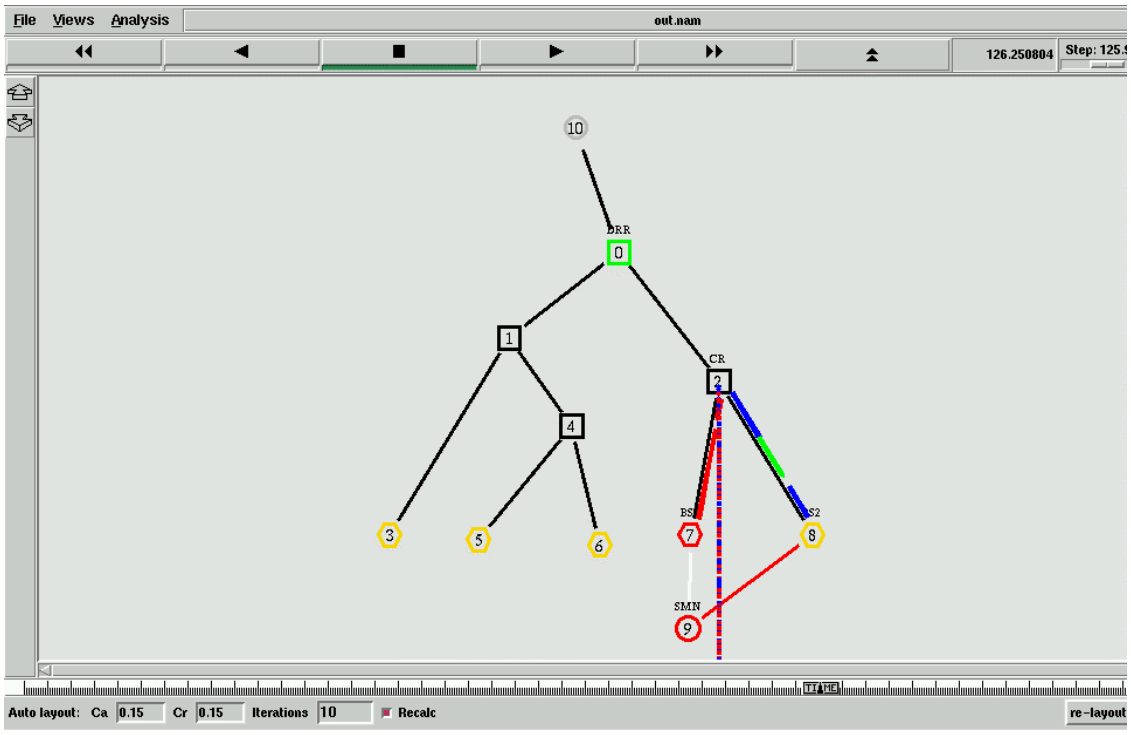


Figure A7-38: Snapshot Just after Handover

Figure A1.44.2-5 shows the moment after the handover. Note that the old wireless link is down and the new active base station is node 8. Green voice traffic follows now the new route via 10-0-2-8-9 to mobile node.

A7.4.3.3 Cases Performance.

In this section we will show the performance of HAWAII and RSVP when de-coupled and loosely-coupled. For both cases we reserve a fixed quantity of bandwidth for RSVP signalling messages. We add a WFQ for RSVP messages with a certain rate to the link to avoid RSVP message loss. The simple formula *n**s*8/30 (*n* is the number of sessions which are going to traverse the link and *s* is the expected average message size) should yield a good approximation of the necessary bandwidth. This value will have to be higher for example if frequent reservation changes occur. Considering a message size value close to 100 bytes and 30 sessions per cell we obtain 750 bps on the wireless link. For the core fixed part we assume that we will hold up to 60 sessions from different cells which sums up to 1500 bps. These values will have to be higher if frequent reservation changes occur.

In order to understand completely the figures we must take into account that there is a planned handover at 100 seconds.

A7.4.3.3.1 De-coupled case.

This case shows the performance of Hawaii and RSVP when both protocols are completely unaware of each other.

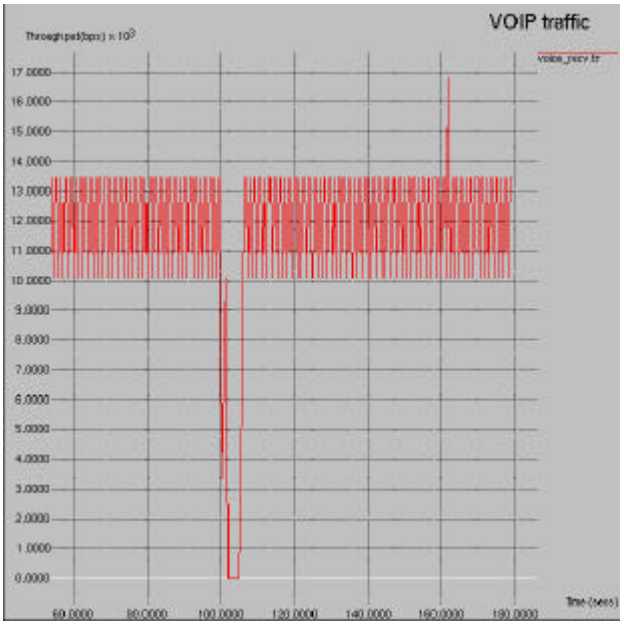


Figure A7-39: Throughput of VoIP Traffic when De-coupled

This graphic shows the throughput of VoIP traffic when de-coupled. When the handover is performed at 100 seconds, the new route only has a reservation to the crossover route, and the interference traffic through the new path, which is much greater in bps than VoIP traffic, prevents VoIP packets to arrive to the mobile host, so it is necessary to wait until the reservation is established to recover the traffic. Approximately at 105 seconds a new reservation is established through the new path so VoIP packets may arrive again to the mobile node and the throughput comes again to its sustained rate.

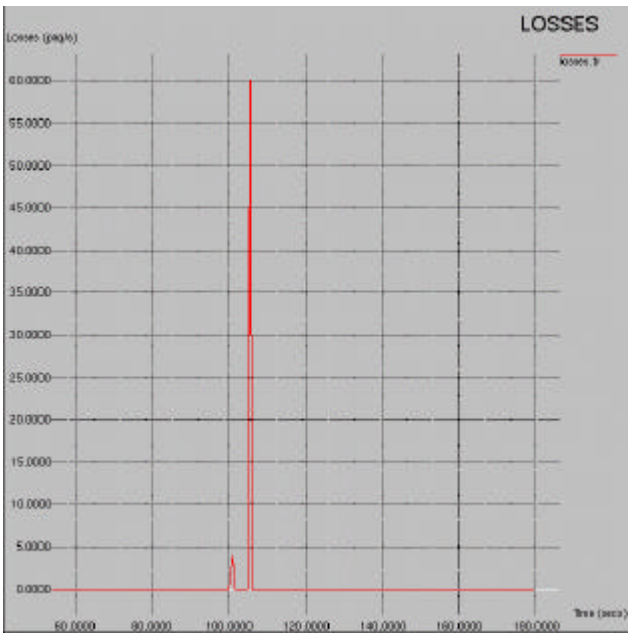


Figure A7-40: Packets of VoIP Traffic Lost per Second when De-coupled

As we can see VoIP packets are lost during handover until the reservation through the new path is established. Note that loss graphs here are measures in packets lost per second and they are not accumulated. After handover up to 60 packets are lost which means that the call is seriously disrupted. The absence of packet loss between the two peaks is a result of the VoIP pattern: there is no traffic in that precisely moment so it is not lost.

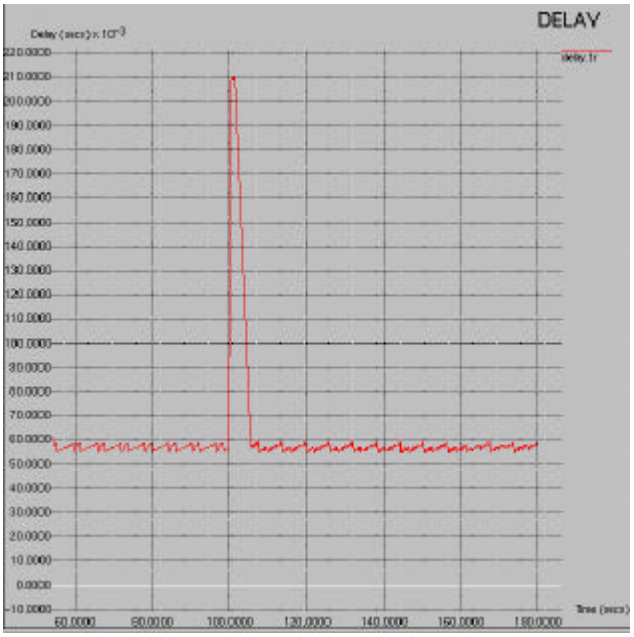


Figure A7-41: Delay of VoIP Traffic when De-coupled

As a consequence of the handover VoIP packets that are not lost suffer a great delay during a long period. Topology is simple so the cause of this delay is just the same as above: the absence of reservation once the new path is established The interference traffic rate is much higher than VoIP rate and the link is saturated, so best effort queue is full. Packets suffer a delay proportional to the length of the queue and some of them, as we have previously seen, are discarded.

A7.4.3.3.2 Loosely Coupled Case

This case is similar to case A with the only difference that HAWAII and RSVP protocols are loosely coupled. We have designed a mechanism to couple both protocols so as they can exchange information

during handoffs. Just after the new route is established the RSVP agent is informed and a refresh of the reservation is sent immediately.

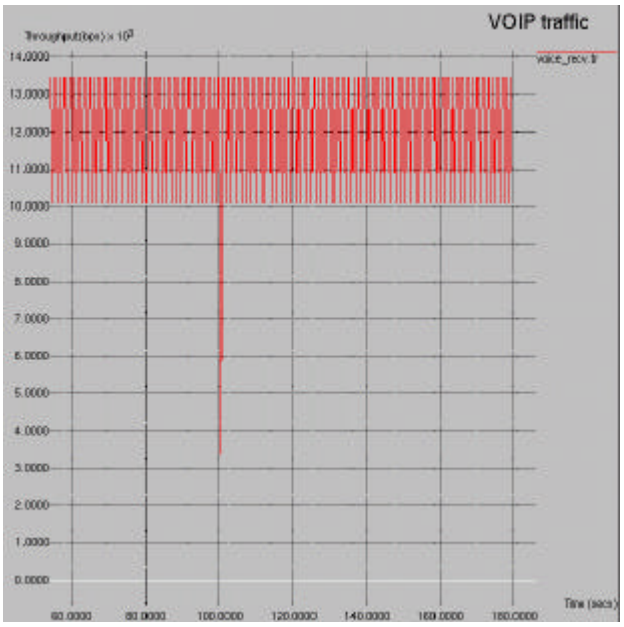


Figure A7-42: Throughput of VoIP traffic when Coupled

Figure A7-42 confirms our thesis. Throughput of VoIP traffic is affected by handover at 100 secs but is much more sustained that when de-coupled. We have to take a look into the following graphs just to see why.

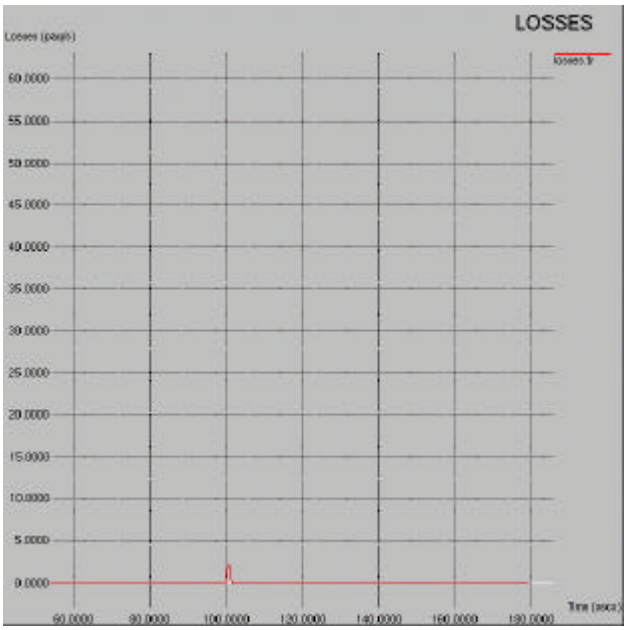


Figure A7-43: Packets of VoIP traffic Lost per Second when Coupled

Figure A7-43 shows that packet loss during handover is minimized. Down to 3-4 packets are lost, mainly due to the proper handover (note the scale when comparing to the loss of the de-coupled case). The rest of the loss caused by the absence of reservation is eliminated just because RSVP refresh messages are sent as soon as the new route is established so the impact of the interference traffic is minimum.

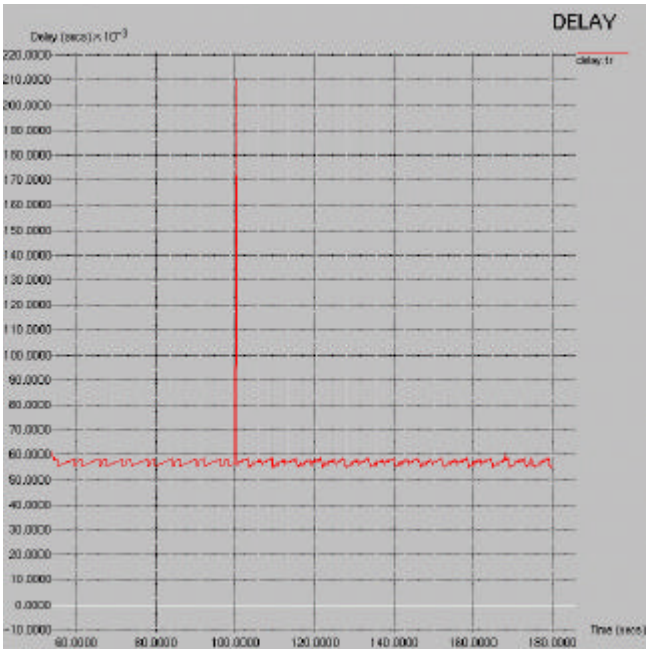


Figure A7-44: Delay of VoIP Traffic when Coupled

Another expected consequence of the coupling is that the impact of handover in delay is also reduced. Some VoIP packets are delayed but only during a few milliseconds, recovering the average of 55-60 milliseconds almost immediately.

A7.4.3.4 Conclusions

As we can extract from the simulation results, the coupling of HAWAII and RSVP offer clear advantages in certain environments. Although it cannot improve handover itself it allows the installation of reservations as soon as the new path to mobile node is stable. This effect is specially interesting in environments as the one showed, when interference traffic would make a lot of VoIP traffic get discarded. We have to note here that these advantages occur because we have a small reservation for the RSVP signalling traffic; just enough for not being discarded as well as the VoIP traffic. If not the time for the new reservation to be installed would increase significantly. This feature is also proposed as an enhancement for the QoS architecture in BRAIN.

In conclusion, the coupling of the micro-mobility protocol and the reservation protocol proposed in the project is easy to perform (see section A4.4.2.1) and the advantages, at least in the loosely coupled case for per-host micro-mobility protocols and when combined to pre-reservation for QoS signalling, are enough to justify it.

A7.5 References

[A7.1] IST-1999-10050/KCL/WP2/MD/001/a1, WP2 Activity Work plan.

[A7.2] IST-1999-100050 project BRAIN, Deliverable D2.I, "BRAIN Access Network Requirements Specification and Evaluation of Current Architectures and Technologies and their Requirements: Core Network and Air Interface", August 2000.

[A7.3] ns website, http://www.isi.edu/nsnam/ .

[A7.4] Keshav S. "REAL: a network simulator", Tech. Rep. 88/472, University of California, Berkeley, December 1988.

[A7.5] Dupuy A, Schwartz J, Yemini, Y., and Bacon, D. "NEST: A network simulation and prototyping testbed", Communications of the ACM 33(10) pp. 64-74, October 1990.

[A7.6] Desbrandes F, Bertolotti S, Dunand L. "OPNET 2.4: an environment for communication network modeling and simulation", in Proceedings of the European Simulation Symposium. Society for Computer Simulation, pp. 609-614, October 1993.

[A7.7] Mah B A. "INSANE Users Manual", The Tenet Group Computer Science Division, University of California, Berkeley 94720, May 1996.
http://HTTP.CS.Berkeley.EDU/~bmah/Software/Insane/InsaneMan.ps

[A7.8] Bagrodia RL, Liao WT. "Maisie: A language for the design of efficient discrete-event simulations", IEEE Transactions on Software Engineering 20(4), pp. 225-238, April 1994.

[A7.9] Perumalla K, Fujimota R, Ogielski A. "TED – A Language for Modelling Telecommunication Networks", ACM SIGMETRICS Performance Evaluation Review 25, 4, March 1998.

[A7.10] Kevin F. "ns Notes and Documentation", The VINT Project, University of California, Berkeley, February 2000.http://www.isi.edu/nsnam/ns/ns-documentation.html

[A7.11] Ahn JS, Danzig P, Estrin D, Timmerman B. "Hybrid technique for simulating high bandwidth delay computer networks", in Proceedings of the ACM SIGMETRICS, pp. 260-261, May 1993.

[A7.12] Brakmo L, Peterson L. "Experiences with network simulation", in Proceedings of the ACM SIGMETRICS, 1996.

[A7.13] Canne S, Xu Y, Yu H. "Network visualization with the VINT network animator nam", Tech. Rep., University of Southern California, pp. 99-703, March 1999.

[A7.14] ns Notes and Documentation contains a description of the interface to the interpreter using Otcl, http://www.isi.edu/nsnam/ns/ns-documentation.html

[A7.15] Diffserv support, http://www.teltec.dcu.ie/~murphys/ns-work/diffserv/index.html

[A7.16] Diffserv module distributed by Nortel, http://www7.nortel.com:8080/CTL/#software

[A7.17] RSVP support, http://www.teltec.dcu.ie/~murphys/ns-work/rsvp/index.html

[A7.18] Insignia support, http://comet.columbia.edu/insignia/ns_source_code.html

[A7.19] Wireless extensions, http://www.monarch.cs.cmu.edu/cmu-ns.html

[A7.20] Mobile IP support, http://www.iprg.nokia.com/~charliep/mobins2/

[A7.21] Mobile IP support developed by University of Southern California,
http://www.icsi.berkeley.edu/~widmer/mnav/ns-extension/

[A7.22] HAWAII support for ns, http://www.tik.ee.ethz.ch/~gfa/

[A7.23] J. Manner, M. Kojo & K. Raatikainen, "SeaWind: A Network Simulator to Study Protocol behaviour in Future Wireless Networks",

[A7.24] IST-1999-100050 project BRAIN, Deliverable D1.1, "Scenarios for mobile IP services and resulting requirements in different wireless networks", August 2000.

[A7.25] C. Keszei, J. Manner, Z. Turanyi and A. Valgo, "Mobility Management and QoS in BRAIN Access Networks", 1st BRAIN Workshop, Kings College London, London, UK, November 2000.