Content-based Video Communication: Methodology and Applications

Paul Bocheck

Submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

in the Graduate School of Arts and Sciences

Columbia University

2000

ABSTRACT

Content-based Video Communication: Methodology and Applications

Paul Bocheck

Most traditional video communication systems consider videos as large files or long bit streams, ignoring the underlying visual content. We introduce a new content-aware framework that explores the strong correlation between video content, resource (bit rate), and utility (quality). Content-based video communication is a promising approach that seamlessly integrates video compression and networking technologies aimed at overcoming current limitations of multimedia communication technology.

Under the new framework, the video traffic and utility functions can be modeled and predicted by analyzing the video content. The resulting traffic and utility function models facilitate joint adaptation between different video streams under dynamic content changes, network conditions, and heterogeneous device capabilities. Utility functions can be used in selecting the optimal transcoding architecture in a pervasive computing environment.

We demonstrate the advantages of the content-aware approach in two applications. First, content-aware models were developed for predicting video traffic for live video streams. The video traffic models were evaluated in a dynamic network resource allocation system. Our simulations have shown that, compared to existing techniques, a significant reduction ($\sim 55\%$ to 70%) in required network resources or up to a 60% reduction in renegotiation frequency can be achieved. Second, we have used the content-aware principle for automatic generation of utility function (sub-

jective quality vs. bit rate) for live video. Our results indicate that high accuracy in estimating utility functions can be achieved.

The main objective of MPEG-7 Universal Multimedia Access is to enable adaptive transport and delivery of multimedia to various client devices with limited communication, processing, storage and display capabilities. Based on our proposals, the media object scalability in the form of utility functions has been included in description schemes for Universal Multimedia Access (UMA) of MPEG-7. The content-aware approach can be directly used for generation of UMA descriptors.

Contents

List of Figures

List of Tables

List of Abbreviations

AMPS	Advanced Mobile Phone System
AP	activity period
AR	autoregressive model
ATM	Asynchronous Transfer Mode
BER	bit error rate
C_i	content class
CAC	call admission control
CA/C	content analyzer / classifier
CB	content-based
CBCC model	content-based content-clustered model
CBR	constant bit rate
CBRC	content-based resource-clustered model
CDV	cell delay variation
CS	content-based scalability
$\mathcal{D}_C$	activity period content descriptor
$\mathcal{D}_G$	global descriptor
$\mathcal{D}_O$	object descriptor
$\mathcal{D}_T$	activity period traffic descriptor
$\mathcal{D}_T, C_i$	characteristics traffic descriptor of content class C_i
$\mathcal{D}_T, T_i$	characteristics traffic descriptor of traffic class T_i
DCT	Discrete Cosine Transform
DRA	dynamic resource allocation
DRS	dynamic rate shaping
D-BIND	resource-bounding interval-dependent video traffic model
GoP	Group of Pictures
GSM	Global System for Mobile communications
MA	media adaptation

MOS	mean opinion score
MPEG	Moving Picture Experts Group
MSE	mean square error
MTA	media traffic agent
PDA	personal digital assistant
PDC	Personal Digital Cellular
PSNR	peak signal-to-noise ratio
QoS	quality of service
RAB	resource allocation broker
RCBR	renegotiated CBR
RM_ACK	resource management acknowledgement
RM_REQ	resource management request
RR	resource reservation
RVBR	renegotiated VBR
SNR	signal-to-noise ratio
T_i	traffic class
TA	traffic analyzer
TDMA	Time Division Multiple Access
TES	Transform-Expand-Sample
TP	traffic predictor
UBR	unspecified bit rate
UF	utility function
UMA	Universal Multimedia Access
VBR	variable bit rate
VC	video content
VO	video object
VOP	video object plane
WAP	wireless application protocol
WSNR	weighted SNR

Chapter 1

Introduction

1.1 Background and Motivation

Over the past decades, processing speed, chip density, memory size and network bandwidth have been exponentially increasing. Continuously falling prices of personal computers and other digital devices influenced their widespread usage and acceptance. Today, personal computers, personal digital assistants (PDA), smartphones and even pagers are powerful enough to send, receive and process the information in many different media. In this context, multimedia refers to the combined use of several media such as text, audio, graphics, video, etc.

Multimedia communications is a burgeoning field [1]. Until now, because of insufficient network bandwidth or inadequate bandwidth management, access to multimedia has been limited primarily to local systems. Use of multimedia in communications is a natural extension of traditional communication technologies such as the telegraph, telephone, fax, etc. The recent burst of Internet applications is a good example of how multimedia-enabling technologies transformed a packet-switched network infrastructure into a global world-wide network with multimedia features [2].

Contrary to the circuit switched telephone networks, future multimedia networks

will be able to transport more efficiently not only the voice or data, but also the video traffic. These networks will also allow easy deployment of new multimedia services. However, compared to data services, many multimedia applications require a quality of service (QoS) guarantee. For example, video and audio applications may impose requirements on bandwidth, end-to-end delay and jitter.

Although today the Internet uses a substantial portion of its bandwidth for multimedia traffic (i.e., image, video and voice), it cannot meet QoS requirements. New advanced Internet protocols (such as RTP, RSVP, IntServ, DiffServ, etc.) [98, 3, 100, 4, 5] allowing limited quality of service guarantees are being proposed and built into the applications and commercial network hardware. In contrast, Asynchronous Transfer Mode (ATM) is a network technology that includes QoS guarantees as a part of its design objectives.

Wireless networks seem to follow a pattern of evolution resembling that of wired networks. In the United States, the first generation of wireless networks was built on top of an analog mobile telephone standard, called Advanced Mobile Phone System (AMPS), that supports voice services only. Besides voice, second generation wireless networks allow various narrow-band data services including the Internet connection, short e-mail (e.g., SMS), etc. However, since second generation wireless networks were originally optimized for voice traffic, their ability to provide multimedia services is limited.

The common denominator of all second generation wireless networks (e.g., GSM, TDMA, PDC, cdmaOne, etc.) is the concept of circuit-switched channels over the wireless interface. It results in limited efficiency of bandwidth usage. Third generation wireless networks will be based on the packet switching technology. In addition, these networks will allow access to up to 2Mb/s - at least 40 times higher than the currently available bandwidth [23]. Thus, third generation wireless networks will

provide high speed access to the Internet and various multimedia applications. Similar to wired networks, a major technical challenge for wireless networks is bandwidth and quality of service management. Given different types of networks, it is essential to support a wide range of QoS.

We focus on quality of service issues for networked video services. Quality of the video, as perceived by the end-user, depends on the entire end-to-end connection that spans (i) source, (ii) network, and (iii) receiver. The network component may constitute multiple hops over wired and wireless segments.

In the source, three components directly affect characteristics of the video traffic. The first component is related to the video content. Video programs have different presentation styles, story structures, lengths, etc. The second component is related to the encoding algorithm. Video programs can be encoded using various standards (i.e., H.263, MPEG-1, MPEG-2, etc.) [90, 91]. The third component is related to the encoding mode. For example, video programs can be encoded as constant-bit rate (CBR) or variable-bit rate (VBR) sources. Other factors on the client side may also influence traffic from the server. For example, video transport and resulting traffic may be constrained by capabilities of the client receiver. Applications with strict timing requirements may pose stringent constraints on the video server [87].

Network bandwidth limitation and variability pose the stringent requirements on bandwidth management. Because of physical limitations (e.g., in wireless networks), it may not be possible to guarantee bandwidth for individual connections. However, it is highly desirable that available bandwidth be distributed among applications in a fair manner.

On the receiver's side, there is a growing trend toward using various heterogeneous devices, e.g., laptops, PDA's, and smartphones equipped with thin wireless application protocol (WAP) browsers. These pervasive computing devices have sig-

nificantly different capabilities (e.g., processing speed, screen size, color capability, limited battery power, etc.) compared to capabilities of desktop computers or workstations. Limited computing power also results in slower decoding of video streams and stricter timing requirements. Battery powered devices will be required to manage the energy efficiently, e.g., by intelligently processing the most important and valuable information for the user. In addition, handheld mobile devices will not have access to the same bandwidth as the desktop does.

Compared to data transport, complex characteristics and heterogeneity of multimedia sources, networks and receivers make the task of network resource management and quality optimization very complex. The development of multimedia networks that support a wide range of services poses significant technical and conceptual challenges.

1.2 Challenges of Multimedia Communications

1.2.1 Video Traffic Modeling

In network engineering, multimedia is defined in terms of network traffic. The characteristics of multimedia traffic depend on many factors. Various forms of multimedia (e.g., voice, still image, video, etc.) are compressed by different standards. For high quality video, MPEG-2 provides good results [91]. However, for VHS quality, MPEG-1 suffices [90]. For low bit-rate streaming applications, H.261, and H.263 codecs and in some cases their proprietary variants are currently used. MPEG-4 is a new standard that allows flexible integration of video, graphics and sound [92, 17]. It also enables content-based (CB) interactive access and a wide range of bit rates. In addition, compared to MPEG-2, new error resilience features that support various network channel conditions are added and are complementary to error correction

features of transport protocols.

Video traffic can be classified into two categories: pre-generated (i.e., generated off-line) and real-time (e.g., live video programs). Pre-generated streams are prepared and encoded prior to the transmission and stored at the server. Many complex issues related to the transmission of real-time video can be avoided by using pre-generated video. For example, its traffic profile, traffic model and descriptors can be accurately computed in advance before the transmission. For real-time video however, exact traffic profiles are, in general, difficult to compute. It is important to note that video traffic does not only depend on the compression algorithms. It is strongly connected to the change of the video content, which is the main source of non-homogeneity in video traffic. Our proposed approach, called content-based modeling, explores the strong correlation between video content and video traffic.

1.2.2 Network Resource Allocation

Video programs can be encoded as constant-bit rate (CBR) or variable-bit rate (VBR) streams. Variable bit rate video encoding is typically used to maintain near constant video quality. Complex bandwidth characteristics of VBR traffic may prevent efficient network utilization (as compared to CBR traffic) in current access networks or wireless networks that has limited bandwidth availability [101]. This trend likely will not disappear, because deployment of next generation networks will not necessarily mean unlimited access or free bandwidth. Without proper resource management schemes and resource reservation protocols, the bandwidth would be soon exhausted.

New techniques that support dynamic bandwidth allocation (DRA) have been proposed to address this issue [99, 28, 24, 25, 26, 27]. Contrary to traditional resource allocation schemes that reserve network resources only at the beginning

of the session, DRA allows reservation (increase and decrease) during the single session. Controlled access to the network resources in DRA is translated into QoS and formulated in terms of probability of source blocking. Prediction of the real-time video traffic and renegotiation scheme poses a challenging issue and is one of the foci of this work.

1.2.3 Content Scalability and Adaptation

The quality of multimedia delivered to the user terminals depends on QoS supported along the entire end-to-end connection. For some applications, it may not be possible or economically viable to support hard end-to-end QoS. These conditions may arise in the Internet or wireless networks. Historically, the Internet was not designed to support QoS. The RSVP protocol [3], although partly able to support QoS, does not scale well to be used in the core Internet network. New initiative that is based on traffic differentiation, DiffServ, may improve quality for some applications, however QoS for individual sources will not be explicitly supported [5].

Similar situation is in wireless networks. Since the wireless networks infrastructure requires substantial investments, network efficiency is a very important factor. However, network efficiency and QoS support for multimedia are conflicting requirements. Although it is, in principle, possible to support bandwidth reservation over the wireless networks, in practice, bandwidth variations over the wireless may substantially limit network utilization and effectiveness.

Given the probabilistic nature of bandwidth variations over the Internet and wireless links, these networks may be able to provide soft QoS only [72]. It will be difficult to support hard QoS for all streams unless the network efficiency is degraded. Based on different application needs, it may be necessary to differentiate QoS services and regulate their usage by established pricing schemes [21]. Under

these conditions, one of the promising solutions is to explore the intrinsic nature of media: scalability.

In general, media adaptation can be supported at the source, receiver, and the network. At the source, multimedia applications may want to adapt the content based on network conditions. At the receiver, media adaptation is concerned with presentation issues [22]. In addition, media scaling can also be supported inside some network nodes (e.g., wireless base stations) in the form of media adaptation service. Combined use of the media adaptation and resource allocation techniques may improve the utilization of the network resources while providing the acceptable quality of the multimedia presentation.

1.3 Thesis Outline

The research theme in this thesis is concerned with the framework of the content-aware video communications integrating studies in communications, video coding, and content analysis. Until recently, these fields were considered relatively independent of each other. Many video traffic models used for prediction conceived video streams as time series generated by unknown stochastic models, parameters of which have to be obtained. Yet, the underlying visual content of the video stream contains vast amount of information that can be used to predict the bit-rate or quality more accurately. In the content-aware framework, this information is extracted by analyzing the video content. Exploration of the correlation between the video content and required resources represents the main idea of the proposed content-aware video communication framework.

The content-aware principle has many applications. In dynamic resource allocation, we will show its effective use for real-time video traffic prediction. Alternatively, it can be used for automatic utility function generation for video content adaptation.

We have used it in selecting the optimal media transcoding operations and content filtering in a pervasive computing environment. Based on our proposal, the media object scalability in the form of utility functions has been included in description schemes for Universal Multimedia Access (UMA) in MPEG-7 [94, 93].

In Chapter 2, we present a generic model of video. In this model, the video is characterized from two perspectives: content and traffic. At the high level of abstraction, video is represented by a hierarchical object structure. From the content point of view, video programs can be classified into different categories, such as the movie, documentaries or sitcoms. Each video program is represented chronologically as a sequence of scenes, shots, and activity periods. An activity period is defined as an elementary, content-homogeneous segment of video program during which optical transition remains relatively unchanged. Activity period is described by content features. Our hypothesis is that the homogeneous visual features in an activity period determine the associated traffic characteristics.

We present hierarchical spatial segmentation scheme for content characterization that is based on video object hierarchy. At different layers of the hierarchy, object regions may or may not correspond to natural video objects in the scene. Individual video objects are characterized by a set of content features.

We present the methods for detection activity periods and associated object features from compressed video stream. We also discuss the implementation of the content analyzer for MPEG-2 videos. At final part of this section, after a brief discussion of traffic modeling, we focus our discussion on characterization of traffic at activity periods. In our simulations, based on conceptual MPEG-2 traffic model, we demonstrate that traffic at activity periods can be characterized by relatively simple stochastic models (e.g., AR(1), random walk models). This observation is one of the important features of the proposed video content modeling. We review the

D-BIND traffic model [56] that will be used for network simulations of content-based video traffic model.

In Chapter 3, we introduce a framework of content-based video traffic modeling. We present the separation principle to model the relationship between video content and video traffic. In addition, we present two content-based approaches for modeling of activity periods: content-based content-clustered (CBCC) and content-based traffic-clustered (CBTC) models. Both models are verified in our experiments. The CBCC model can be used for synthetic traffic generation. The CBTC model is used for traffic prediction in dynamic resource allocation (DRA), described in Chapter 4.

In Chapter 4, we discuss resource allocation issues in bandwidth-limited networks. In particular, we present a content-based dynamic resource allocation system that combines the above content-based traffic models and several DRA schemes. In our experiments, we have implemented a real-time content analyzer that is based on a fully automated analysis of compressed videos. The MPEG-2 content analyzer consists of four modules that are invoked sequentially. In the analyzer, components for activity period detection, video object detection and content feature estimation are included.

Performance study of content-based DRA was based on trace-driven simulations. Results were obtained using a single 54000-frame-long trace (30 minutes) of an MPEG-2 encoded movie. Network simulations revealed that the content-based approach achieved better performance (in terms of link utilization) than other existing schemes (RVBR, RCBR) [54, 53]. The link utilization achieved by RVBR was substantially less than utilization achieved by CBTC scheme (about 55% - 70% difference) [75].

Utility functions represent a powerful framework for characterizing the ability of applications to adapt to varying network conditions. Specifically, in the context of

bandwidth allocation, utility functions indicate a media object's quality as a function of available bandwidth. In Chapter 5, we describe conceptual model of multimedia communications based on utility-based adaptation. The aim of the model is to encompass, in general, source and network characteristics, terminal capabilities and user preference into the unified framework.

In particular, we present one of the components of the model: the content-based utility estimator. The estimator implements an automated technique to estimate utility functions for video objects. The content-based principle is applied and machine learning techniques are used. The system uses video content, represented by a small set of content features, to determine the utility class of an object. Because video features can be automatically extracted from compressed video streams, this technique is also suitable for real-time applications.

Accuracy of MPEG-2 and MPEG-4 content-based utility estimators was evaluated in simulations. For the experiment using MPEG-2 video (17 utility classes), the classification accuracy of the whole set of utility functions was 91%. Similarly, high classification accuracy of 80% - 85% was achieved using MPEG-4 traces [95].

In Chapter 6, we discuss some key issues concerning content-aware communications. In particular, we present the work toward the content description for universal multimedia access (UMA) [94] that was included as part of MPEG-7. MPEG-7 is an ongoing standardization process headed by Moving Picture's Experts Group (MPEG) [92]. One of the goals of the proposed standard is description of the multimedia material. Some of the applications of MPEG-7 and UMA will deal with access, delivery and presentation issues of multi-resolution audio-video objects. Based on the results presented in this thesis, a set of content descriptors for UMA applicable for media adaptation and transcoding was incorporated into the MPEG-7 UMA.

1.4 Thesis Contributions

(a) We present a generic model of video applicable to traffic modeling.

(b) We present a hierarchical spatial segmentation scheme content characterization of activity periods.

(c) We define content features and present methods for their extraction in the compressed-domain.

(d) For our simulations, we implemented a real-time MPEG-2 content analyzer.

(e) We performed network simulations of conceptual MPEG-2 video traffic model. In this model, the traffic was characterized at time-scale of activity periods. The results show that traffic within activity periods can be modeled by relatively simple stochastic models.

(f) We formulate the separation principle as a basis for content-based video traffic modeling.

(g) We propose two variants of the content-based video traffic model: content-clustered (CBCC) and traffic clustered (CBTC). The accuracy of both models was evaluated and compared.

(h) We propose a system model for the content-based dynamic resource allocation. The system was evaluated in network simulations and compared to other dynamic resource allocation systems.

(i) We present a conceptual model of multimedia communications based on utility-based adaptation.

(j) We propose a system model for accelerated generation of utility functions. We present results based on two variants of the model for subjective and

objective quality metrics.

(k) We present a content description framework for Universal Multimedia Access (UMA) that was included into the current MPEG-7 standard.

Chapter 2

Video Characterization

2.1 Introduction

The compressed video stream is produced by a video encoder. The main goal of video encoding is the reduction in the bandwidth of the original video source. Traditionally, video traffic modeling focused on finding best-fit parameters of well-known stochastic models (e.g., AR, ARIMA, MMPP, etc. [101, 102, 103]). The model parameters were estimated based on empirical traces of real video programs. Accuracy of traffic models can be found by comparing traffic characteristics (e.g., first or second order statistics, histogram, etc.) or results of network simulations. In many cases, parameters of traffic models were obtained under the assumption that experimental traces are homogeneous for the duration of the entire video program. In addition, in most of these models, no extraneous information about the content source was used.

The encoded video stream contains the information necessary for reconstructing the video program, i.e., its entire video content. As we will show in Chapter 3, there is a high correlation between the video content and the video traffic. This important observation suggests that the video content can be used for video traffic modeling. We argue that if this content information can be extracted in an adequate way, it can

be used to boost the performance and accuracy of the traffic prediction models. In addition, traffic models based on video content allow generation of synthetic traffic for different video categories (e.g., movie, news, etc.). In this chapter, we propose the hierarchical scheme for the content and traffic description of video programs. In the rest of the thesis, video sequence and video are used to refer to video program.

2.2 Video Hierarchy

In general, video programs differ in their content, presentation style, purpose, etc. The structure of video programs (e.g., the particular sequence of scenes and shots) might differ considerably accross various video categories. Movie, documentary, sports, and news are examples of different video categories. The video categories differ in editing methods, scene and shot length characteristics, types of camera operations, use of special trick modes, etc. Many factors determine the structure of the video programs. The psychological factors are among the most important. The video program should present the story in an understandable way. The other factors may be the editor's personal preferences and experience, length limitations, available footage, and editing technology, etc. Although humans can easily determine the video category, automatic classification by machines is rather complicated. Automatic classification can be based on statistical analysis or knowledge-based analysis of video content.

The video program can be represented as a sequence of scenes. A scene is defined as a continuous video segment (i.e., time interval) that can be described by some meaningful element such as location or action. For example, a scene may depict a dramatic episode at one common location. Scenes can be further divided into one or more shots. A shot is defined as a continuous video segment that is outlined by an abrupt optical transition. The abrupt optical transitions are usually a result of

video editing process.

At a high level of abstraction, the chronological sequence of scenes or shots can be visualized by space-time diagrams [6], which depict the correct playback order of shots. The playback order is a result of the particular sequence of typical video editing operations. Several different video editing operations are used in practice: spatial or temporal deletion, flash-back and flash-forward, and parallel and multiple parallel actions. For example, the time-space diagrams of two video programs A and B representing two different video categories are depicted in Figure 2-1. Video A is a movie and video B is a real-time sport program. Shots are shown as horizontal lines. The length of each line corresponds to the shot duration. Note that the playback order of shots is not necessarily in the order in which the shots were acquired. For example, movie A contains flash-back temporal and space transitions between the locations A, B, C, D while real-time sport program B contains space transitions only: four cameras at fixed locations A, B, C, D are used to show the sport event.

It is important to note that during a shot, the continuous optical transition may occur. This situation may arise, for example, during a long camera panning. If the transition results in substantial change of video content, video programs cannot be sufficiently described by a sequence of shots. These changes during the single shot will result in different bit-rate statistics. In other words, representation by space-time diagrams is not adequate for content-based video traffic modeling.

In the context of video traffic modeling, we propose to represent video programs as a sequence of content-homogeneous time intervals, called activity periods (AP). The activity period is defined as a continuous video segment during which optical transition remains relatively unchanged. In other words, activity periods are bound by an observable optical transition such as a change of camera operation, a change in the number of objects, etc. For example, during a single shot, the camera may be

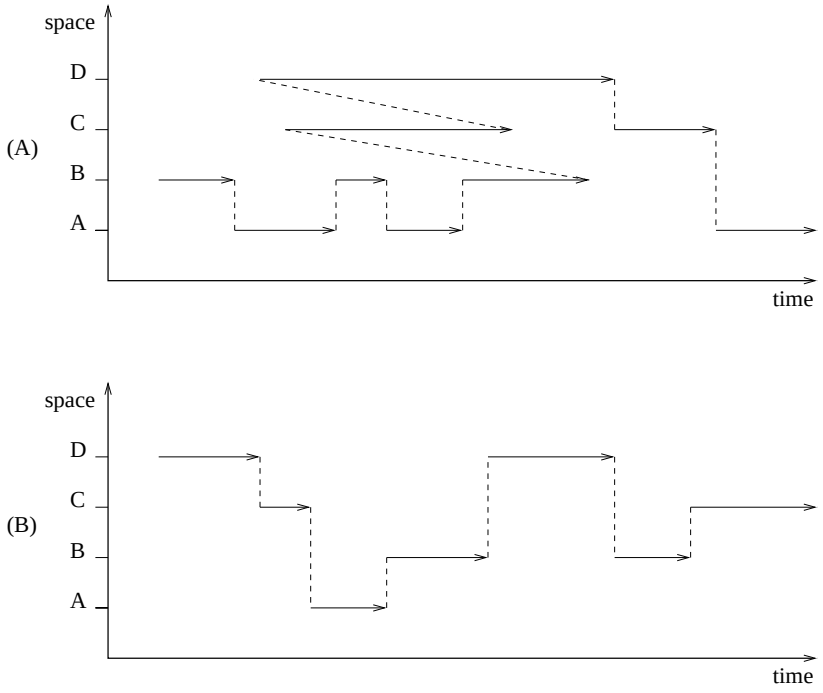

Figure 2-1: Space-time diagram. (A) movie program, (B) real-time sport event program

static (first activity period), then a new object may appear in the picture (second activity period) and finally, the camera may follow the moving object (third activity period). In our model, the activity period represents an elementary unit for content description. The above described hierarchical representation of video programs is depicted in Figure 2-2. Also note that in practice, activity periods consist of a sequence of video frames.

As we will show later, the change of the video content, e.g., background change, can induce a significant change of the bit-rate. These changes in the traffic characteristics typically correspond to boundaries of scenes, shots or activity periods. Changes can assume variations of abrupt or smooth transitions. Changes between activity periods are more difficult to detect if there is a contiguous optical transition between two activity periods.

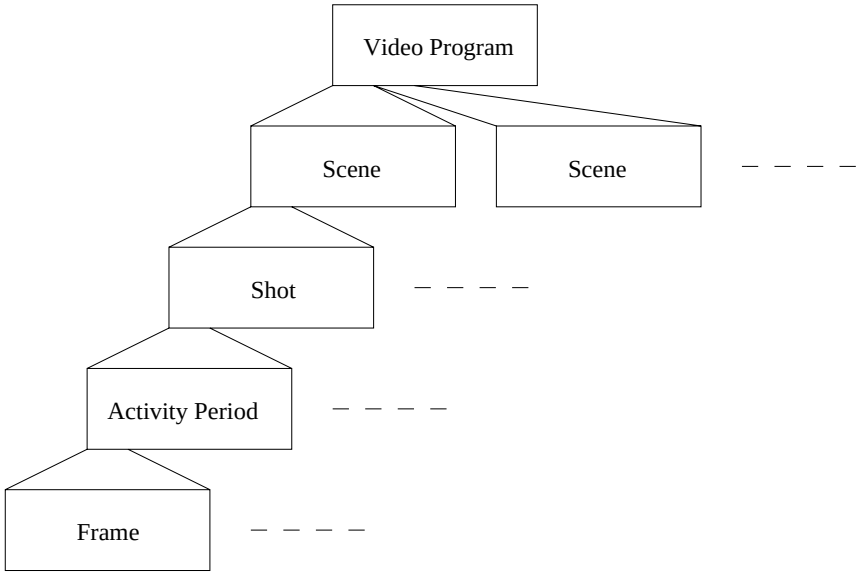

Figure 2-2: Hierarchical decomposition of video programs

In the context of traffic modeling, activity periods are characterized by duration, content and traffic. The content of activity periods is described in Chapter 2.3 and the traffic of activity periods is described in Chapter 2.4.

2.3 Content Characterization

2.3.1 Spatial Decomposition

In order to describe the video content, first, we present the following hierarchical spatial segmentation scheme. Figure 2-3 depicts a single video frame and its decomposition. At the root of the segmentation tree (first decomposition layer), the entire frame is considered as a single object region. At other decomposition layers, the video frame is segmented into several object regions. For example, at the second layer, two object regions are shown. The first object region, o_{21}, is associated with a collection of foreground objects (car, house, tree, etc.). The second object region, o_{22}, is associated with the background (sky).

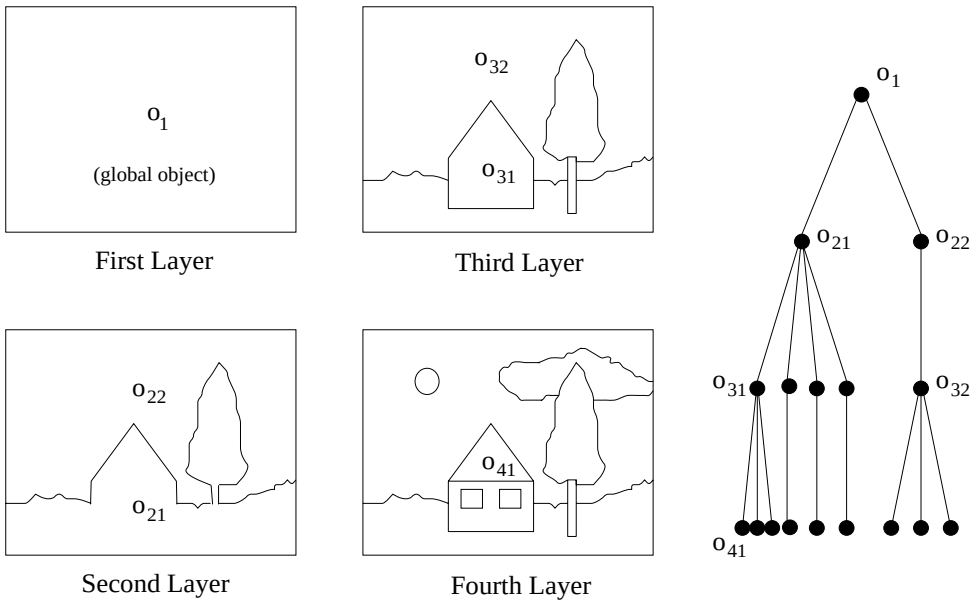

Figure 2-3: Hierarchical spatial segmentation.

Based on our previous example, object regions may directly relate to real objects or just their parts. With new advanced image segmentation algorithms, it is possible to roughly segment object regions in real time* [58, 60, 61, 62, 63].

In practice, the algorithms for spatial segmentation are relatively complex and require a substantial amount of processing power. Many image techniques, suitable for automatic spatial segmentation of video sequences, are currently available [62, 63, 104]. Although some techniques may not be able to match segmented object regions and real objects perfectly, they have been applied successfully to video coding and indexing [58, 64]. Unfortunately, many of these techniques are not designed for real-time operations. Alternative approaches are segmentation algorithms that operate on video streams directly in the compressed domain [59]. The main advantage of compressed-domain algorithms is their higher speed when applied to already

*With the accuracy adequate for traffic modeling since the perfect correspondence with real-world objects is still difficult.

compressed streams as compared to non-compressed domain algorithms. Because of the commonly used block-based structure of compression algorithms (e.g., MPEG), compressed-domain segmentation algorithms suffer from a lower resolution in segmentation boundaries. However, for the purpose of video traffic modeling, extraction of detailed object regions and their accurate correspondence to physical objects is not necessary. In our work, automatic real-time compressed-domain segmentation is used for object region detection and spatial segmentation.

The object region can be conceptualized as a snapshot of the video object (VO), defined in both spatial and temporal domains. In this sense, the outlined hierarchical spatial decomposition is compatible with the video object definition in MPEG-4. The object region corresponds directly to the MPEG-4 video object plane. Video object (e.g., the sequence of object regions over the activity period) corresponds directly to MPEG-4 video object.

Note that the object segmentation tools do not recognize the semantic information in the scene. However, this does not invalidate our definition of video object, since, for our purpose, we are interested in the perceived video object rather than the real one.

2.3.2 Content Description

The content description model, depicted in Figure 2-4, is synergetic with the hierarchical spatial segmentation. In the following, the model is used to describe the video content in terms of objects and their descriptors. The model is flexible in terms of segmentation accuracy, number of content features, number of layers, etc.

The activity period, δ, containing a set of video objects, o_i, is described by a single global descriptor, $\mathcal{D}_G(\delta)$, and several object descriptors, $\mathcal{D}_O(o_i)$. Each descriptor contains a set of content features. In general, the content features characterize (i)

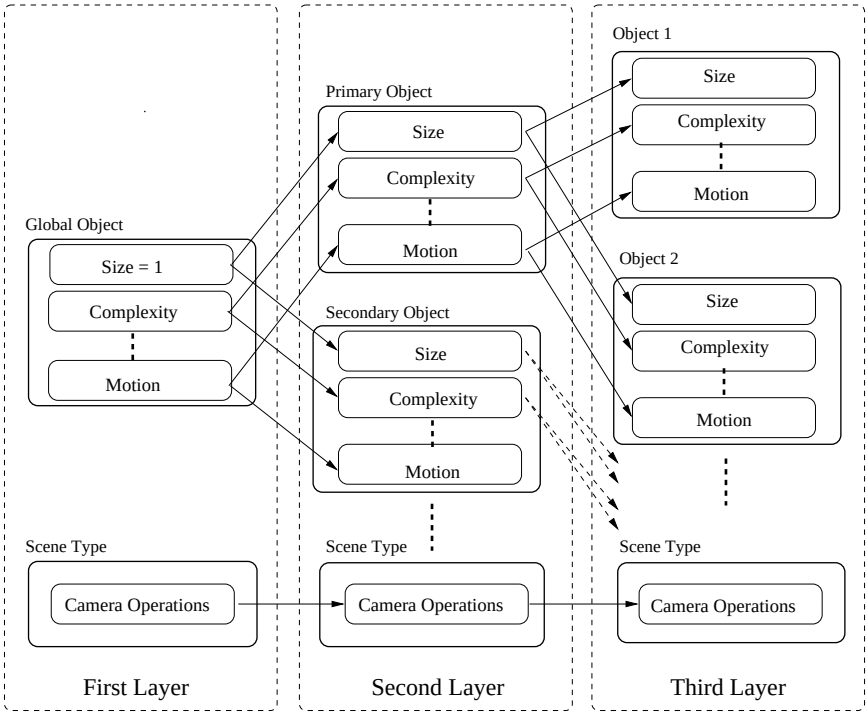

Figure 2-4: Content description model.

the entire visual scene or (ii) individual video objects. The features that relate to all objects in the activity period are called global features, $\mathcal{F}_G$, and assigned to $\mathcal{D}_G$. The content features that describe individual objects only are called object features, $\mathcal{F}_O$, and assigned to $\mathcal{D}_O$.

The above content description model is quite general. In practice, it is desirable to keep the number of content features on a manageable level. Hence, only features that are important to the given application will be selected. In video traffic modeling, the content features that influence the bit-rate will be chosen. For example, an important feature affecting the video bit-rate is the type of camera operation employed (i.e., static, panning, zooming, etc.).

The individual video object is characterized by a set of object features, $\mathcal{F}_O$. Each $\mathcal{F}_O$ represents a particular content feature of the video object. For example, video objects can have different sizes, shapes, complexities (e.g., cluttered vs. smooth

texture), and can move at various speeds and in different directions. Again, selection of particular features depends on application and complexity constraints. In the context of video traffic modeling, some object features might have a direct influence on the bit-rate (e.g., spatial complexity or motion) while others might not (e.g., motion direction).

The entire activity period is then described by a content descriptor D_C:

$$\mathcal{D}_C(\delta) \triangleq \{\mathcal{D}_G(\delta), \mathcal{D}_O(o_1), \mathcal{D}_O(o_2), \cdots, \mathcal{D}_O(o_i), \cdots \mathcal{D}_O(o_N)\} \tag{2.1}$$

$$\mathcal{D}_G(\cdot) \triangleq \{\mathcal{F}_{G1}, \mathcal{F}_{G2}, \cdots, \mathcal{F}_{GJ}\} \tag{2.2}$$

$$\mathcal{D}_O(\cdot) \triangleq \{\mathcal{F}_{O1}, \mathcal{F}_{O2}, \cdots, \mathcal{F}_{OK}\} \tag{2.3}$$

where N is a number of objects in activity period, J is the number of global features, $\mathcal{F}_G$, and K is a number of object features, $\mathcal{F}_O$.

2.3.3 Content Features

The content features can be defined in either uncompressed or compressed domains. Estimation of content features in the compressed domain has several advantages. In practice, if the video is already encoded, feature estimation in the compressed domain reduces computation time because frames do not need to be converted back to the original uncompressed domain. The disadvantage of defining the content features in the compressed domain may be the dependency on particular compression mechanisms. However, such a close relationship with the encoder structure may be worthwhile in real-time applications. Fully automated analysis of compressed video signals has shown great promise [59, 65] and can provide a satisfactory approximation for the purpose of content-based video traffic modeling.

Sometimes, the video content information may be supplied directly by a digi-

tal video camera and associated scripts. Additional external information that can be used for describing video content includes scene cut schedules, storyboards describing scene activities and camera operations. Future cameras might also provide supplemental information about zooming, panning and other features related to the video content. In the future, the content information may be available in the form of MPEG-7 descriptors and description schemes.

In MPEG-2, video object size, spatial complexity and object motion can be obtained directly from the compressed video stream. These features are defined in the following way.

Denote N the total number of transform blocks in the frame. Size $S(o_j)$ of video object is defined as the number of DCT blocks b_i belonging fully to the video object (e.g. excluding blocks that belong to the object partially only):

$$S(o_j) \triangleq \sum_{k\,=\,0}^{N-1} 1_{\{b_k \in\, o_j\}} \tag{2.4}$$

In the compressed domain, object motion magnitude, $M(o_j)$, is estimated from motion vectors of blocks belonging to the same object. For MPEG video, motion vectors can be extracted directly from compressed streams. In this case, motion, $M(o_j)$, can be estimated as follows:

$$M(o_j) \triangleq \frac{1}{S(o_j)} \sum_{b_k \in\, o_j} |\vec{m}_k| \tag{2.5}$$

where $\vec{m}_k$ is a motion vector of the block b_k. When motion information is not present in the compressed stream (i.e., during I-frames), video object motion can be predicted from previous frames or evaluated in the spatial domain. However, this approach may require full decoding of compressed streams resulting in increased computational delay.

We define the spatial complexity of a video object in relation to its entropy: as a number of bits needed for that object, given entropy-coded uniform quantizer. In block-based compression algorithms, each block is independently encoded. Under this assumption, spatial complexity, $C(o_j)$, of the object o_j can be estimated as the sum of a number of bits B_k of blocks belonging to the same video object:

$$C(o_j) \triangleq \frac{1}{S(o_j)} \sum_{b_k \in \ o_j} B_k \tag{2.6}$$

Because of the independence of transform (frequency) coefficients, the number of bits for each block, $B_k = \sum_{i \in b_k} y(i)$, can be expressed simply as the sum of bits, $y(i)$, of all of its coefficients i. Under the assumption that coefficient i can be modeled as a discrete i.i.d. process with zero mean and variance σ^2 and under the criterion of small to medium distortion (compared to standard deviation), the number of bits $y(i)$ can be estimated as [66]:

$$y(i) = \frac{1}{2} \log_2 (12 \ \epsilon^2 \ \frac{\sigma^2}{\Delta^2}) \tag{2.7}$$

where Δ is a quantizer step size and ϵ is a model-dependent constant corresponding to the frequency coefficient, which is equal to about 1, 1.2, and 1.4 for Uniform, Laplacian and Gaussian distribution (pdf) respectively.

Then, the number of bits, B_k, of the MxM block, b_k size would be:

$$B_k = \frac{1}{M^2} \sum_{i=0}^{M^2-1} y(i) = \frac{1}{2M^2} \log_2 \prod_{i=0}^{M^2-1} (12\epsilon_i^2 \frac{\sigma_i^2}{\Delta_i^2}) \tag{2.8}$$

where $y(i)$ denotes number of bits for coefficient i.

In practice, complexity can be estimated in two ways. In the compressed domain, DCT coefficients are already run-length encoded (e.g., I-frame in MPEG-2). In that

case, complexity of the object can be estimated from a number of bits used for its DCT coefficients. If the motion estimation prevents this direct estimation (i.e., encoded coefficients represent residual error of motion compensation in P and B frames in MPEG), the complexity can be estimated from Equation 2.7, measuring the variance of the frequency components. The disadvantage of this method is that it needs to partially decode the video stream to the transform domains. If a delay in estimation of entropy is allowed (i.e., until the next I-frame), it may be sufficient to estimate the complexity using the former approach only.

2.3.4 MPEG-2 Content Analyzer

In our implementation, the MPEG-2 automatic content analyzer consists of three modules that are invoked sequentially. In the first module, the activity periods are detected. In the second module, video objects are detected in each activity period. Finally, activity period content descriptors are estimated.

The real-time automatic content analyzer was based on Columbia University's MPEG-2 decoder [59]. Since the content analyzer process streams directly in the compressed domain, the decoder was simplified to contain only parts that are necessary for activity period detection, video object detection and content feature estimation. In particular, the computationally intensive inverse DCT function was fully omitted. In our implementation, the complexity was estimated from I-frames only and motion was estimated from P-frames only. This simplification resulted in real-time performance on a general-purpose workstation. On SUN SPARCstation 5 it was possible to analyze each video frame for its content in less then 10 ms, i.e., before the next frame came in.

2.3.4.1 Detection of Camera Operation

Global motion and camera operations are determined from the compressed video stream with the use of motion vectors. Although motion vectors are estimated in the encoding process for each macroblock independently and do not necessarily represent true motion, they can be successfully used to estimate global motion, camera operations and to detect moving objects in P- and B-frames.

The size of macroblocks (16×16 pixels) in MPEG-2 limits the accuracy of motion detection and object extraction. For that reason, it is sufficient to use relatively simple 2D motion model for global motion estimation. Global motion directly corresponds to camera operations.

In general, global motion can be found, given motion vectors of all macroblocks in the frame, by using the histogram segmentation method introduced in [59]. The method is based on detecting of the single dominant motion direction from the histogram of motion vector angles. If such a dominant motion can be found, the camera operation is declared as pure panning with parameters corresponding to the average magnitude of motion vectors corresponding to the "peak" bin of the histogram. If no dominant motions can be found and the average magnitude of motion vectors is close to zero, the camera operation is declared as static. Otherwise, the camera operation is declared as zooming. In that case, zooming and panning parameters are estimated using the least squares method.

2.3.4.2 Detection of Video Objects

Moving video objects are found by recovering their non-zero local motions. This is accomplished by using the global motion compensation method described above. After global motion compenstation (e.g., subtraction of global motion from all macroblocks in the frame), macroblocks that belong to the background will have mag-

nitude of motion vectors close to zero. On the other hand, macroblocks belonging to moving objects have non-zero motion vectors. After global motion compensation and noise filtering, moving object boundaries are found by a histogram segmentation method. Macroblocks that belong to the bin with maximum number of macroblocks are marked as belonging to the same object. Although several video objects moving in different directions can be found, the histogram segmentation method was used to recover one dominant moving object only. After the macroblock labeling operation, video objects are extracted using a simple block merging operation that is used to delete objects consisting of single macroblock and merge regions surrounded by labeled macroblocks, e.g. containing "holes".

The following content features have been chosen to characterize video object's content (i.e., $\mathcal{D}_O$): object size, spatial complexity, and motion were estimated as defined in Equations 2.4, 2.5, and 2.6 respectively.

2.4 Traffic within Activity Periods

Despite a large number of VBR models proposed over the last decade, no model appears to be suitable for all different types of real-world video traffic [30]. VBR video exhibits complex characteristics (e.g., self-similarity, variable time-scale property, long-term dependency and non-stationarity) that make model identification and estimation very hard. In general, there is a tradeoff between model complexity and accuracy. To simplify model complexity, it is sometimes necessary to accept various assumptions that result in decreased accuracy. The final selection of a suitable model depends on the application and its requirements.

Fundamentally, the stochastic process representing the video traffic is non-stationary. Some consider the traffic model as doubly stochastic, i.e. parameters of the model are itself random variables. The video input can be seen as an external source de-

fined by entropy. Output from the video encoder can also be modeled as entropy of the source after the decorrelative procedure is applied (e.g. spatial and temporal correlation is extracted).

The removal of such spatial and temporal correlation is a general objective of compression algorithms. However, the correlation at larger time-scales (in the order of several frame periods) still remains. The correlation at these scales depends on several factors: (i) editing structure of the program (ii) video format and (iii) and encoding algorithm. Video format and encoding mechanism are defined by standards, for example H.263 and MPEG-2.

Because of its natural dependency on the video content, the VBR video traffic is non-stationary. For example, after the scene change, the stream generated by the encoder might completely change its stochastic characteristics. Furthermore, it was found that video traffic exhibits self-similar properties and long-term dependency. The behavior of real video sources is rather complex and cannot be accurately modeled by simple stochastic models; more complex doubly stochastic models are typically not analytically tractable and their parameters are difficult to obtain. However, depending on applications, many of these undesired statistical characteristics can be disregarded under the assumption that traffic behavior is modeled at a relatively short time-scale, compared to the length of the video program. For example, the video traffic can be modeled in any of the following time-scales: scene, shot, and activity period (Figure 2-5) [38]. When long sequences of non-homogeneous video programs are segmented into homogeneous activity periods, traffic at each activity period can be modeled by relatively simple stochastic models.

Video traffic at time-scales shorter than activity periods is directly related to the compression mechanism or network architecture. For example, MPEG-2 video traffic can be modeled at the level of a group of pictures (GoP), frames, in slices and in

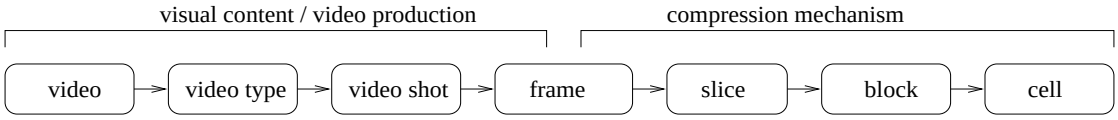

Figure 2-5: Frame/block-based video decomposition.

block domains. Given Asynchronous Transfer Mode (ATM) networking technology, video traffic can be modeled in terms of inter-arrival times of fixed-size ATM cells (53-byte packets). Models in these time-scales are considered to be compression mechanism or networking architecture related.

2.4.1 Traffic Models

Traffic models are used for different purposes. Stochastic models are generally used in an analytical estimation of the queuing performance; effective bandwidth and resource bounding models are used for the admission control. Resource bounding models are described in terms of bounds on the source traffic. Autoregressive models are typically used for the real-time traffic prediction in dynamic resource allocation. Encoder-specific synthetic traffic generators are used as a traffic source for simulation of buffering requirements and cell loss estimation at the network multiplexer.

A good overview of traffic modeling can be found in [30]. Typically, traffic models are based on renewal processes (special case Poisson and Bernoulli), Markov-renewal processes, Markov processes, Markov-modulated processes (special case Markov-modulated Poisson Process, fluid traffic models), and autoregressive models (AR, MA, ARMA, ARIMA) [101, 102]. These stochastic models can capture mean, variance or autocorrelation function of video sequences. Their parameters can be found offline or approximated online by measurement-based algorithms. The Transform-Expand-Sample (TES) models can match both marginal distributions and autocorrelation of original traces. The main disadvantage of TES modeling is its relatively

complex process of finding parameters that rely heavily on heuristic computer-aided tools [31]. Other features such as self-similarity, found in local area networks, and long range dependency can be modeled by doubly-stochastic processes.

The concept of effective bandwidth is used in call admission. Estimation of the effective bandwidth is based on the theory of large deviations. The approximation to the cell loss probability holds only in the asymptotic sense under the limiting assumptions of a very large number of multiplexed sources and very large network buffers. In the asymptotic case, for very large number of sources and large buffers, effective bandwidth of homogeneous statistically multiplexed sources is equal to the sum of their individual effective bandwidths. In that sense, effective bandwidth, which is less than peak rate, assigned to the stream is a measure of its resource usage. In order to keep the required QoS, the call admission algorithm can accept only so many sources as long as the sum of the individual effective bandwidth does not exceed the link capacity.

Analytical derivations of effective bandwidth are known only for very few homogeneous traffic models such as on-off or Markov chain stochastic processes [55, 7]. Although it is also possible to estimate the effective bandwidth on-line using the real-time traffic measurements [8], it is very hard to monitor the source conformance to its effective bandwidth in real-time.

Two traffic models for video phone sessions were proposed by Maglaris et. al. [32]: the continuous-state autoregressive Markov model, and the discrete-state continuous time Markov process. The first model can be used in simulations, but it does not lead to simple results of queuing analysis. The second model can be used to analyze performance of a statistical multiplexer. This work was later extended [33] by considering scenes with multiple activity levels. The extended 2D Markov process model included both short-term and long-term correlation.

Skelly et. al. [34] introduced a single source model based on an arrival rate histogram of smoothed traffic. In this model, queuing performance of aggregated traffic in the ATM multiplexer was predicted simply by convolution. Authors concluded that this model is quite general, in that it does not assume any specific cell arrival process. The histogram should be estimated over the smoothed traffic with the rate of modulation much smaller than that of the cell arrival process. Based on simulation results, authors argue that while long-term correlation of video traffic is important, its actual form is not. The proposed model does not attempt to characterize the arrival process in the time domain, but only its important factors (e.g., bit-rate distribution) that can be directly used for prediction of network resources required for video stream.

The class of traffic models called scene models is able to capture scene changes, different motion activity within scenes, etc. For example, the model for variety of motion activities within scenes was proposed by Yegenoglu [36]. Their model has three motion activity classes corresponding to low, medium and high motion. Traffic generated by each frame is represented by AR(1) process belonging to one of three motion classes. Each process has different parameters. The discrete-time Markov process governs changes between motion activity classes.

Leduc et. al. provided a comprehensive statistical analysis of real VBR sources [38]. They developed different traffic models from the ATM cell domain up to the TV program domain and validated their theoretical approach with a 25-hour-long television program. They argue that each domain has to be modeled by fundamentally different stochastic models. Video scene changes and the mean bit-rate within scenes were identified as the most important sources of non-stationarity observed within TV programs.

Ramamurthy et. al. [35] developed a VBR video model for many different video

sources ranging from a videophone to a full motion video with scene changes. They observed that video traffic depends on a variety of factors, including: video content, amount of movement in the scene, and type of coding technique used. Their model consists of the sum of two independent AR(1) processes and Markov-modulated normally distributed random process. The later models increase of the bit-rate during scene changes and sum of AR(1) processes capture autocorrelation function.

Another model that accounts for scene changes was proposed by Lazar [37]. In their work, VBR video sequences were modeled as a collection of stationary subsequences corresponding to scenes. Within the individual scene, the TES model was used to match both marginal distribution and autocorrelation. Based on their data, they established that a scene length can be modeled as the Bernoulli process.

The previously described stochastic models did not consider unique features of the encoding algorithms. The encoder-specific models are used to model specific encoders, such as MPEG-2, H.261, H.263, etc. In addition to frame ordering, these models typically consider correlation between different frame types. The statistical analysis and modeling of MPEG-2 VBR traffic was presented by Heyman [39]. Their model explicitly considers the correlation between different frame types within the MPEG-2 standard. It is able to follow the periodic GoP structure such that a synthetic trace, produced by the model, resembles the real MPEG-2 trace very closely. Their model was extended to account for frequent scene changes in [40]. The proposed compound model consists of separate models for scene lengths (Gamma, Weibull, and generalized Pareto distributions), scene-change frames and intra-scene frames (discrete autoregressive DAR(1) process). They concluded that although the modeling approach is the same for all 11 experimental sequences, the use of a single model with few parameters that would be applicable to all sequences does not seem to be possible.

Another, Non-Markovian model for VBR video sequences was proposed by Frater [41]. Their scenic model is similar to the DAR(1) model except it uses non-exponential scene length distribution.

2.4.2 A Conceptual MPEG-2 Model

In the following, we present a conceptual traffic model for MPEG-2 video. The main purpose of the model is to show that traffic in activity periods can be modeled by simple stochastic models. This is demonstrated by a comparison of the first and second order statistics and ATM network simulation.

The model generates the traffic frame by frame, using two processes for MPEG-2 specific parameters: complexity and motion. In addition, the model preserves the MPEG-2 frame structure. Each activity period δ_i is characterized by descriptor $\mathcal{E}(\delta_\rangle)$ which contains two stochastic processes $R_i = \{R_{i,k};\ k = 1, 2, \cdots\}$ and $M_i = \{M_{i,k};\ k = 1, 2, \cdots\}$:

$$\mathcal{E}(\delta_i) \rightarrow \{R_i,\ M_i\} \tag{2.9}$$

where k denotes a frame number index. We call R_i and M_i activity period reference processes. R_i models the complexity parameter. M_i models the motion parameter. Both parameters depend on the MPEG-2 coding technique: complexity is defined as the sum of the absolute value of DCT coefficients; motion is defined as the sum of the absolute value of motion coefficients.

Both R_i and M_i reference models are used to create the MPEG-2 specific frame structure in the following way. First, three values $S_{I,k}$, $S_{P,k}$, and $S_{B,k}$ are generated, each representing the sizes of modeled I, P, and B frames respectively. For simplicity, we assume these values are arranged in the sequences $S_I = \{S_{I,k};\ k = 1, 2, ...\}$,

$S_P = \{S_{P,k};\ k = 1, 2, ...\}$, and $S_B = \{S_{B,k};\ k = 1, 2, ...\}$. Based on the MPEG-2 standard, the video frames are properly assembled from S_I, S_P, and S_B sequences to reflect the correct frame ordering. Assuming a GOP (Group of Pictures) of size 12 with 4 subgroups each starting with I or P reference picture, the frame ordering sequence is as follows:

$$S_{I,1},\ S_{B,2},\ S_{B,3},\ S_{P,4},\ S_{B,5},\ S_{B,6}, \cdots, S_{P,10},\ S_{B,11},\ S_{B,12},\ S_{I,13}, \cdots \tag{2.10}$$

The values of $S_{I,k}$, $S_{P,k}$, and $S_{B,k}$ frames are modeled as follows. Since I frames are intra-frame coded using the DCT transformation, they are directly related to the current frame complexity parameter, denoted as $R_{i,k}$. Then for each frame k, its compressed I-frame size, denoted as $S_{I,k}$, is modeled as:

$$S_{I,k} = f_I(R_{i,k}) = C_I\ R_{i,k} \tag{2.11}$$

where f_I is mapping function simplified to a scaling constant C_I in this case.

On the other hand, P and B frames are coded using the motion compensation. Their frame size is the combination of complexity, $R_{i,k}$, and motion parameters, denoted as $M_{i,k}$. We approximate the size of P and B frames, denoted as $S_{P,k}$ and $S_{B,k}$ respectively for each frame k as:

$$S_{P,k} = f_P(R_{i,k},\ M_{i,k}) = R_{i,k}\ M_{i,k} \tag{2.12}$$

$$S_{B,k} = f_B(S_{P,k},\ M_{i,k}) = S_{P,k}\ M_{i,k} + S_{P,k}\ (1 - M_{i,k})\ \beta \tag{2.13}$$

here f_P and f_B are mapping functions, $R_{i,k}$ is an complexity parameter, and $b = 0.5$ is a empirically estimated scaling coefficient. Equation 2.12 expresses the observed dependency of $S_{P,k}$ frame size on M_k and R_k. It shows that the P-frame size de-

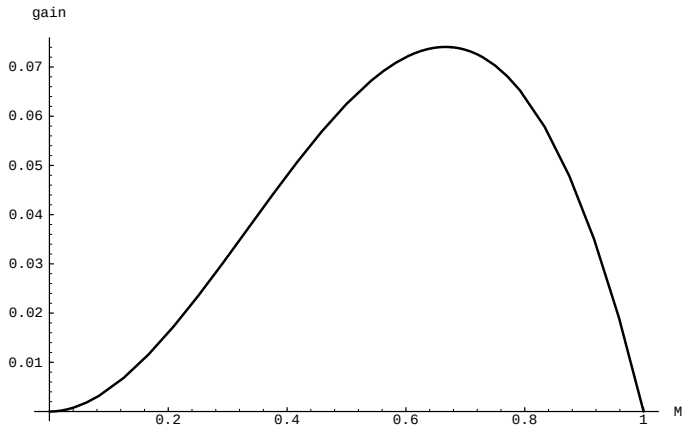

Figure 2-6: Dependency of B-frame of compression gain on M_k.

pends on both complexity and motion parameters. Equation 2.13 expresses the observed nonlinear dependency of $S_{B,k}$ frame size on M_k. The intuition behind this non-linearity is the following: it was observed that for low motion, the advantage of bidirectional motion compensated coding was not significant, while for higher motion, the substantial compression gain was observed. Note however, that for very high motion, the B frame size approaches the P frame size (Figure 2-6).

2.4.2.1 Results

We evaluated the model by comparing the first and second order statistics of the original and modeled trace. To be able to test our model, we created the video content template of the original source sequence-1, depicted on the left of Figure 2-7. For the traffic model at activity periods, we have chosen the random walk for its relative low computation requirements and high correlation between close samples. We approximated the complexity parameter of the first frame, denoted as $R_{i,1}$, and step size, denoted as Δ_R as:

$$R_{i,1} = m_{R_i}, \qquad \Delta^2_{R_i} = \sigma^2_{R_i}/\tau_i \tag{2.14}$$

where m_{R_i} and $\sigma^2_{R_i}$ are the mean and variance of I frame sizes in the activity period i, and τ_i is the activity period length. For the still camera motion we set $\sigma = 0$. For each activity period, motion M_1 and motion step size were evaluated similarly by normalizing each frame size in a GOP with respect to its I frame (the first frame in GOP) and taking the average and variance of such normalized P frames.

The model and real traces are depicted in Figure 2-7. In the model trace we can identify similar periods corresponding to activity periods in the original source. Autocorrelation of I, P and B frames, depicted in Figure 2-8 also resemble the similar behavior. We can confirm its slow decaying characteristic, as reported in [10].

To further evaluate the model, we simulated the ATM multiplexer loaded with several sources, either real or modeled. The results are depicted in Figure 2-9. Four cases of 100, 120, 140 and 200 sources correspond to load r=0.47, 0.57, 0.66, and 0.95. The bit error rate (BER) of the model closely matched the bit error rate of the real source for the low buffer values and all four utilizations. Note that for high multiplexer loads the model estimates the change of the slope of the bit error rate characteristics accurately. Multiple-slope characteristics, similar to those appearing in Figure 2-9, were reported in [11].

2.4.3 D-BIND Video Traffic Model

The D-BIND is a deterministic video traffic model that was introduced by Zhang et. al. [54]. It was used as a traffic model for renegotiated VBR (RVBR) resource allocation service [56]. The D-BIND model has the ability to characterize source burstiness at different time scales, i.e., variable-length time intervals. Time-scale dependent properties were observed in streams generated by VBR video encoders.

The use of the D-BIND model for traffic characterization of activity periods has the following advantages [54]. It has been shown that peak rate allocation is not

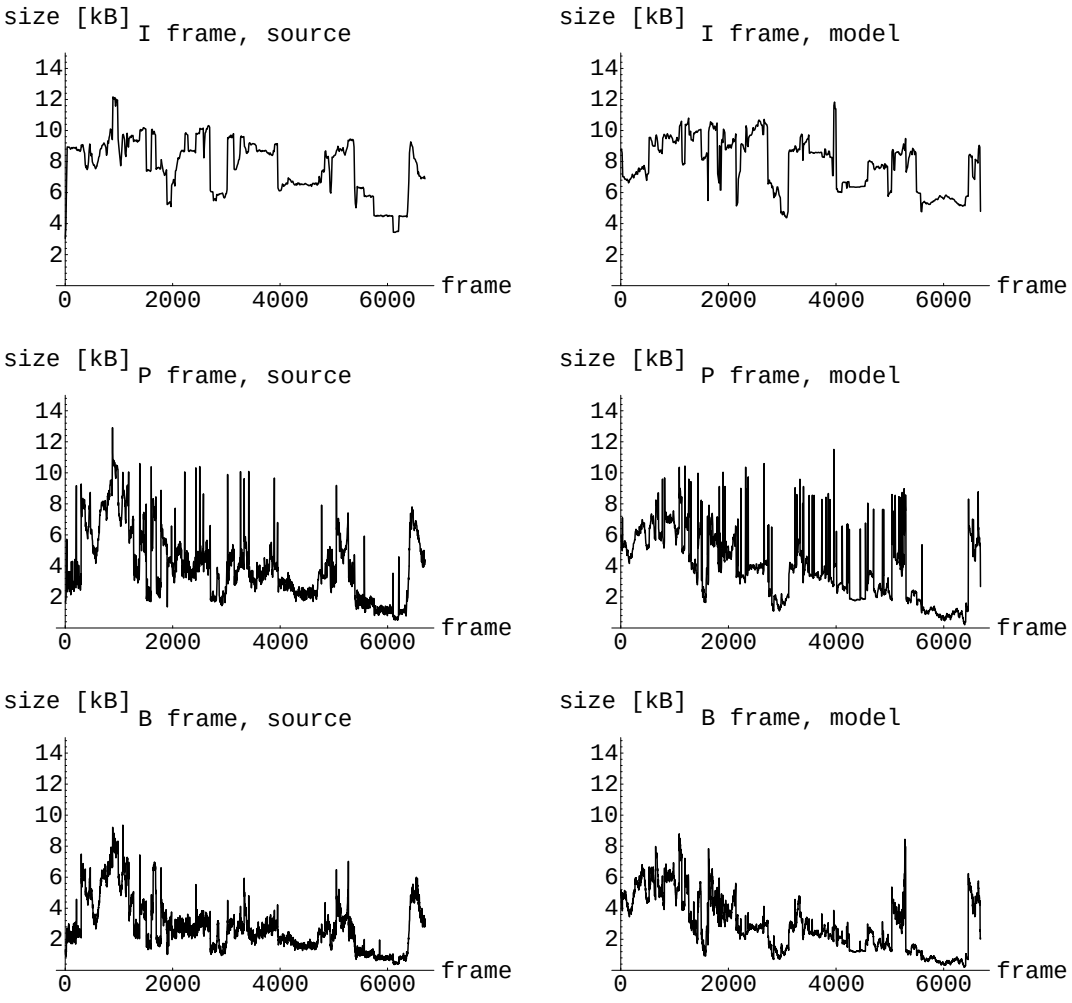

Figure 2-7: Trace of I, P, and B-frames of original and modeled video trace.

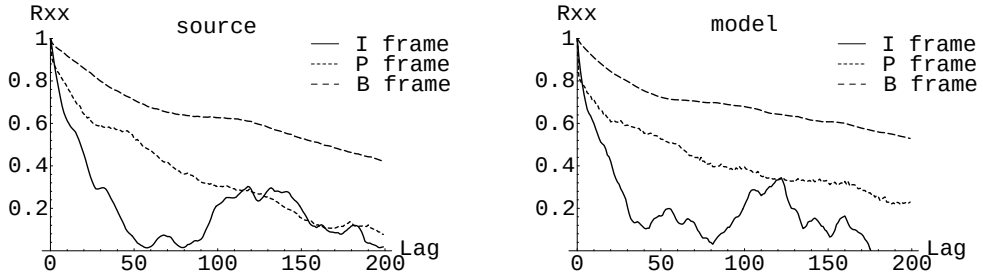

Figure 2-8: Autocorrelation of I, P, and B-frame sequences of source and modeled video trace.

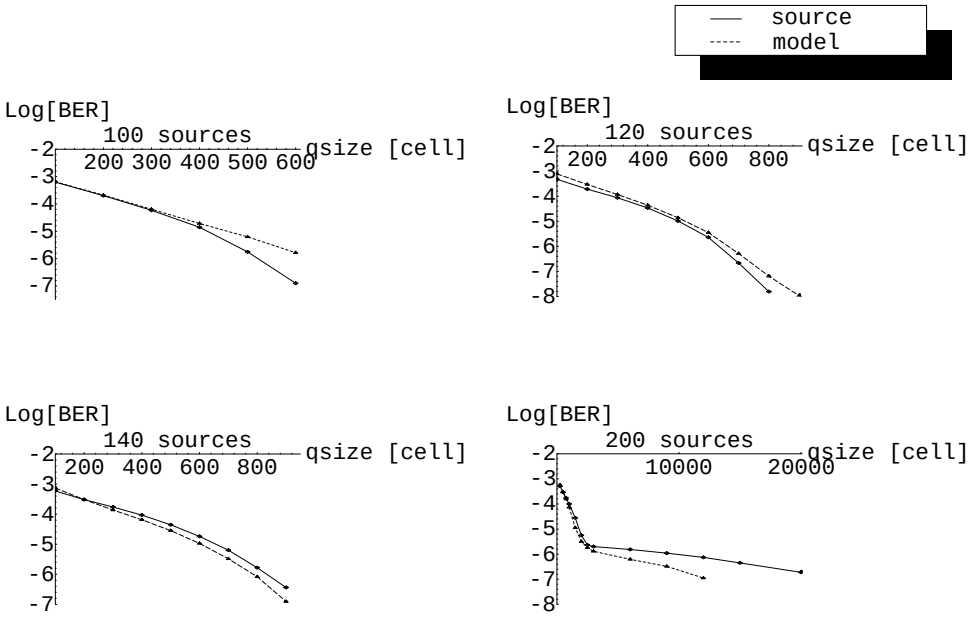

Figure 2-9: Network simulation.

necessary to provide the deterministic QoS guarantee for the VBR traffic. By using the D-BIND traffic descriptor, higher network utilization was achieved. In addition, instead of modeling the process itself, the bounds on the traffic are modeled. The complex encoder-specific structures corresponding to different frame-types (I-, P-, and B-frames in MPEG-2) do not need to be explicitly modeled. This significantly simplifies the model. An advantage of modeling in terms of bounds in the traffic is that it can be efficiently used for resource management during the dynamic resource allocation [56]. The D-BIND traffic descriptor can be directly used for that purpose. Finally, traffic conformity to the deterministic-bound source descriptors can be easily policed using a number of leaky bucket regulators.

The unique feature of the D-BIND traffic model is its ability to capture the time-scale dependent characteristics of the video source. The D-BIND model is defined as follows [56]. Denote $A[\tau, \tau+t]$ cumulative arrivals of a source s during an interval $[\tau, \tau+t]$. Define an empirical envelope $B^*(t)$:

$$B^*(t) \stackrel{\triangle}{=} \sup_{\tau>0} A[\tau, \tau+t] \qquad \forall t > 0 \tag{2.15}$$

The empirical envelope, $B^*(t)$, represents the tightest time-invariant bound on source arrivals for every interval $[\tau, \tau+t]$ of length t. Define a family of traffic-constrained functions $\mathcal{B}$:

$$\mathcal{B} \stackrel{\triangle}{=} \{B(t) \mid B^*(t) \leq B(t)\} \qquad \forall t > 0 \tag{2.16}$$

The source s is deterministically bound by the traffic-constrained function $B(t)$ if $B(t) \in \mathcal{B}$. In other words, for source arrivals during a time interval of length t the

following holds:

$$A[\tau, \tau + t] \leq B^*(t) \leq B(t) \qquad \forall t, \tau > 0 \tag{2.17}$$

The D-BIND model refers to a parameterized continuous traffic-constrained function $B_{W_T}(t) \in \mathcal{B}$ defined on a set of P points $W_T = \{(q_k, t_k) \mid k = 1, 2, ..., P\}$:

$$B_{W_T}(t) \triangleq q_k + \frac{q_k - q_{k-1}}{t_k - t_{k-1}}(t - t_k) \qquad t_{k-1} \leq t \leq t_k \tag{2.18}$$

with the assumption that $q_0 = 0$ and $t_0 = 0$. In other words, a set of points W_T defines $B_{W_T}(t)$ as a continuous piece-wise linear function bounding the empirical envelope $B^*(t)$ described above. We refer to the traffic-constrained function $B_{W_T}(t)$ as a D-BIND traffic-constrained function. The number of time intervals t_k generally depends on required accuracy of the piece-wise linear approximation. In experiments, described in [56], a maximum of seven time intervals t_k in a range from several tenths of ms to several seconds were used for the D-BIND traffic descriptor. In our simulation experiments, we have used nine time intervals t_k ranging from 33 ms (one frame interval) to 2.5 s (corresponding to {1,4,7,10,13,25,37,49,61} frames) for the D-BIND traffic descriptor.

2.4.3.1 D-BIND Descriptor Estimation

There could be different ways to construct $B_{W_T}(t)$ off-line, but the following procedure can be used to construct $B^*_{W_T}(t)$ that represents a tight bound on $B^*(t)$. In general, $B_{W_T}(t)$ is different for different video or activity periods. The procedure computes, for the empirical envelope $B^*(t)$ and a given set of time intervals $T = \{t_k\}_{k=1}^{P}$, values of q_k such that $W_T = \{(q_k, t_k) \mid k = 1, 2, ..., P\}$ defines the D-BIND traffic constrained function $B^*_{W_T}(t)$. The algorithm is as follows:

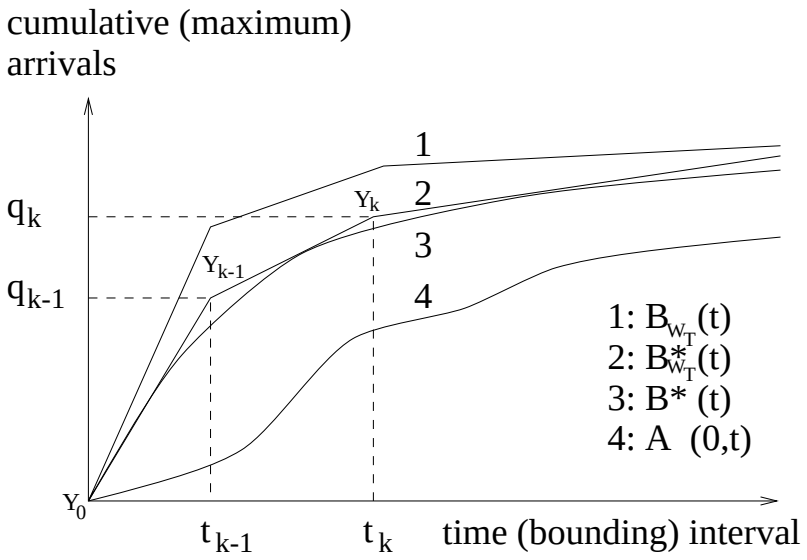

Figure 2-10: $A(0,t)$, $B^*(t)$ and $B_{W_T}(t)$ traffic functions.

1. Input: $B^*(t)$ and given set $T = \{t_k\}_{k=1}^{P}$
2. Initialize starting point $Y_0 = \{q_0 = 0, t_0 = 0\}$
3. For $k = 1$ to P
4. Find $Y_k = \{q_k, t_k\}$ corresponding to minimum q_k such that
   line $\overline{Y_{k-1}, Y_k}$ is never below $B^*(t)$ on time interval (t_{k-1}, t_k).
5. end
6. Output: $W_T = \{Y_k \mid k = 1, 2, ..., P\}$

Figure 2-10 depicts schematically the cumulative arrival function $A(0,t)$, the empirical envelope $B^*(t)$ and two D-BIND traffic constrained functions $B_{W_T}(t)$ and $B^*_{W_T}(t)$. The $B^*_{W_T}(t)$ represents an optimal (tight) D-BIND traffic constrained function. The $B_{W_T}(t)$ is another possible D-BIND traffic constrained function, which is not tight.

It is sometimes more convenient to express the D-BIND traffic constrained func-

tion as rate-interval pairs $R_T = \{(r_k, t_k) \mid k = 1, 2, ..., P\}$ such that $r_k = q_k/t_k$ denotes bounding rate over the interval length t_k. We refer to this specification as the D-BIND traffic resource descriptor.

Chapter 3

Content-based Traffic Modeling

3.1 Introduction

The traffic generated by multimedia and video applications is, in general, non-stationary and bursty with long-term dependence and fractal characteristics. In addition, high bandwidth video streams may consume a substantial portion of available network capacity. Dimensions of networks and complexity of traffic characteristics are the main reasons that make exact analytic evaluation of the network performance practically intractable. Exceptions are asymptotic results of statistical multiplexing derived from fluid models using large deviations theory. These solutions, however, were derived for some simple stochastic models only (e.g., on-off Markovian mini-source) and results have been made explicitly for large queue size and large number of superimposed sources [29]. Characteristics of real world VBR traffic such as non-stationary behavior and periodic frame-type structure of MPEG cannot be adequately expressed by these models.

Because of these complex traffic characteristics, network simulations are in many cases used as an aid in developing and dimensioning future communication networks. The validity of network simulations depends on the accuracy of the traffic model. The traffic modeling is an essential instrument in solving communication issues

connected with bandwidth management, buffering, latency, and quality of service. Traffic models are used in three major areas: as part of the analytical solution to statistical multiplexing in queuing networks, as synthetic traffic generators for network simulations, and as real-time traffic predictors of network resources. The suitability of the traffic model for the network simulation depends on how close the model resembles the characteristics of the real video traffic. Although real traffic generated off-line can still be used in simulations, there is often only a limited number of video traces available. Synthetic traffic generators have the ability to adjust to different types of encoding mechanisms. In general, synthetic video traffic models should be flexible in accommodating a wide range of video traffic encoders and video sources by adjusting only a few parameters.

Another important application of traffic modeling is real-time bandwidth prediction for use in networks supporting dynamic bandwidth allocation. In packet networks, video quality can be substantially reduced when packets are dropped inside the network in an uncontrolled fashion due to the insufficient bandwidth. Because of high burstiness in traffic, static call admission and reservation based on peak rate is not desirable. The effectiveness of dynamic resource allocation (DRA) depends on accuracy of traffic prediction models. In general, traffic prediction models should not be too complex so that video streams can be processed in real-time.

A review of previous works in previous chapter shows that in traditional traffic modeling, video traffic was assumed to be generated by a homogeneous stochastic process, parameters of which have to be estimated. Most of these models did not consider any additional content-related information, i.e., whether this is a conference call with a head-and-shoulder static video scene or a 10-s advertisement of high activity. This information can be extracted from the video stream or can be available from the scene and object descriptors of the future MPEG-7 standard. The models

that can use this information are called content-based.

The motivation behind the use of video content is not surprising because it naturally reflects the video stream encoding process. The close relationship between video content and bit rate was previously discovered in VBR video streams. Rodriguez proposed a new approach for modeling of VBR video sequence [42]. It was based on an assumption that (i) every video sequence can be characterized by a set of fundamental indexes or parameters relating to the video content and (ii) that the bit-rate can be generated by a "parametric model" of the corresponding indexes. They suggest that an appropriate linear model can be obtained for each encoding mechanism. The fundamental indexes that can be estimated from the video source included pixel histogram, spatial and temporal correlation, motion index and other parameters.

The idea of fundamental indexes can be related to content features, described later in Chapter 2. However, as we will show later, besides this similarity, the models differ in several key points. For example, the concept of video object and its content features is not considered in the "parametric model". The features that relate to the visual content are also not used (e.g., motion). In addition, the "parametric model" assumes only a linear combination of parameters. On the other hand, our content-based model is based on video objects and their content features. Furthermore, the classification scheme provides a more flexible mapping between the video content and the bit-rate.

The modeling based on the description of the video content fundamentally deviates from the traditional traffic modeling. The content-based model differs from traditional models in that it considers the video content as a natural source of video traffic non-stationarity. In this sense, the video traffic depends on both the video content and encoder. Consequently, modeling the video traffic by exploring ad-

ditional information about its nature and generating process leads to results that are more realistic and accurate compared to traditional models. The content-based model can be used for generation of synthetic traffic using deterministic or probabilistic content scripts. Additionally, the model can be used for rate control as part of video encoders or for real-time traffic prediction and resource allocation.

3.2 Relationship Between Video Content and Traffic

Figure 3-1 depicts a 1000-frame segment of VBR MPEG-2 encoded video stream. It was created from the movie Forrest Gump [43] using Columbia University's MPEG-2 software encoder. This example illustrates the non-stationary characteristics of the video traffic. To better visualize the trend of I, P, and B frames, frame envelopes were used. The frame envelope connects frames of the same type (see legend). Vertical dotted lines mark changes in video content of video (such as a change of camera view, beginning or ending of camera or object motion, etc.). Sample frames corresponding to each segment in Figure 3-1 are depicted in Figures 3-3 to 3-10 respectively. D-BIND rate constrained functions for each segment are depicted in Figure 3-2.

Comparing Figure 3-1 and Figure 3-2 with Figures 3-3 to 3-10, it can be observed that video trace characteristics (e.g., discontinuities and changes in its stochastic nature) are directly related to the changes and characteristics of the video content. The scene changes coincide with singularity points delimiting the abrupt changes in the stochastic nature of the trace. However, the characteristics remain relatively homogeneous at times between the segments. For example, segment 1 is an image sequence with a smooth texture and high-speed camera panning in a horizontal direction. Similarly, high-speed horizontal camera panning appears at segments 4 and 5, although both of them are not as smooth as segment 1 (e.g., medium smoothness).

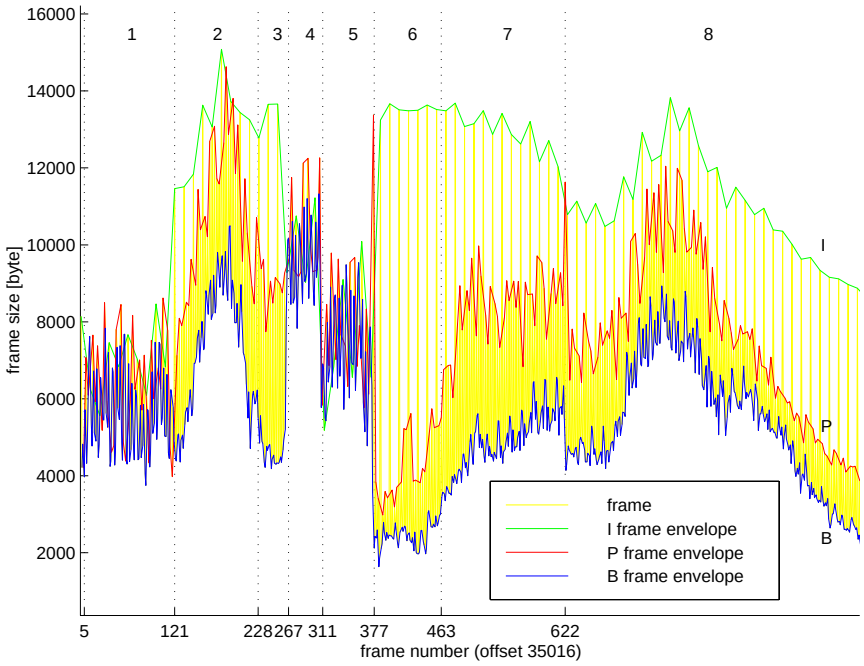

Figure 3-1: MPEG-2 VBR trace of movie Forrest Gump.

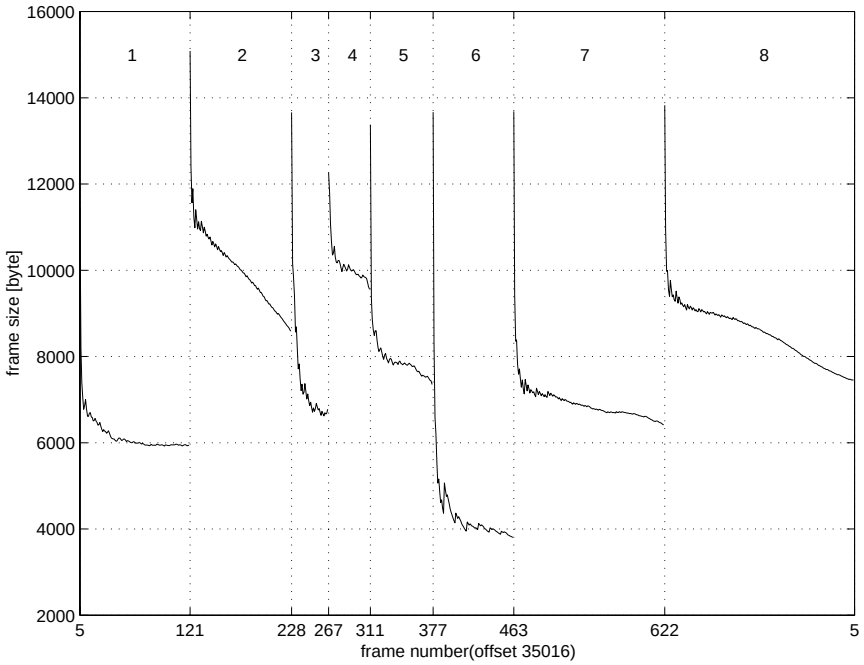

Figure 3-2: B-BIND rate constrained functions.

Segment 2 corresponds to an image sequence with a relatively rough texture and medium speed camera panning along the vertical axis. Segments 3 and 6 correspond to static image sequences without camera motion, but with rough texture; segment 3 corresponds to the image sequence with a single large object moving with medium speed while the images in segment 6 have no moving objects. Both segments 7 and 8 are image sequences with a rough texture background; additionally, segment 7 corresponds to an image sequence with medium speed camera zooming, while segment 8 corresponds to an image sequence with high speed camera zooming.

3.2.1 Separation principle

There are several important observations, marking the importance of the content and its use in the traffic modeling. During the compression, the video content of the video is encoded into the bit stream. It is clear that the video traffic is influenced by two independent factors: video content and the compression mechanism. In essence, based on the same video content, different video encoders can generate traffic streams with different stochastic characteristics. Additionally, it has been shown that traffic characteristics corresponding to periods of homogeneous content (i.e., activity period) do not possess extreme and complicated behaviors (e.g. non-stationarity) and therefore, can be modeled by stationary Markov or AR(n) models [38]. These observations are summarized as the separation principle, representing the important concept of content-based video traffic modeling [57].

The separation principle is schematically shown in Figure 3-11. It depicts the content-based video traffic model comprising (i) content-dependent and (ii) encoder-dependent component. The content-dependent model corresponds to the entire video production (i.e., visual scene composition, editing style, control of the video camera, etc.). The video sequence is represented as a collection of activity peri-

Figure 3-3: Segment 1

Figure 3-4: Segment 5

Figure 3-5: Segment 2

Figure 3-6: Segment 6

Figure 3-7: Segment 3

Figure 3-8: Segment 7

Figure 3-9: Segment 4

Figure 3-10: Segment 8

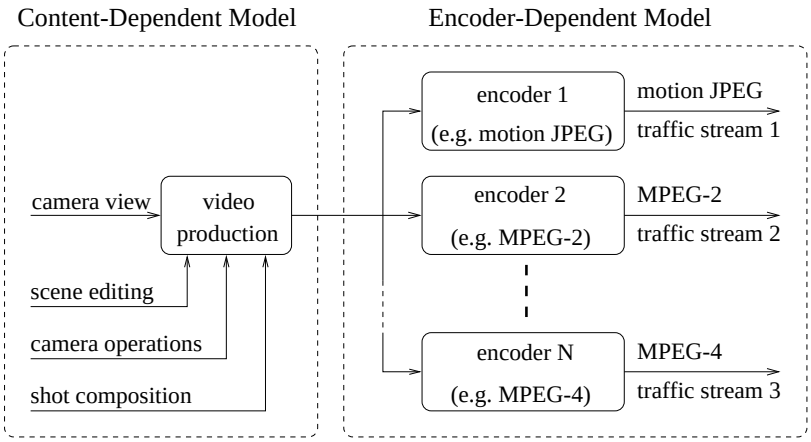

Figure 3-11: Separation principle in the Content-based Video Traffic Model.

ods during which video content is assumed homogeneous. On the other hand, the encoder-dependent component models a compression mechanism. Use of a relatively simple traffic descriptor greatly simplifies the encoder-dependent component. This component captures quasi-stationary traffic behavior within the activity period.

Relative independence between content-dependent and encoder-dependent components of the model does not imply independence of source bit-rate on both models. On the contrary, the bandwidth depends on both the video content and the compression technique.

3.3 CB Content-Clustered Model (CBCC)

The content-clustered model is used to investigate a relationship between video content and its traffic. In practice, the content-clustered model can be used as a synthetic traffic generator.

Assume that a video program is segmented into activity periods, characterized by the content descriptor $\mathcal{D}_C$, as described in Section 3.3.3. Bandwidth requirements

of activity periods are characterized by the traffic model, described by the traffic descriptor $\mathcal{D}_T$. For the purpose of video traffic modeling, it is beneficial that the content features are chosen based on their close relationship to bandwidth requirements. However, the exact form of such relationship is complex. To simplify the model, we propose that each activity period to be classified according to its content features.

The content classification scheme expresses the relationship between the video content and bandwidth requirements of activity periods. Denote $\mathcal{C} = \{C_i \mid i = 1, 2, \cdots, g\}$ a content-based classification scheme consisting of g content classes C_i. Denote $\mathcal{D}_{T,C_i}(\delta_j)$ a characteristic traffic descriptor of activity period δ_j, associated with the content class C_i. Formally, content-based resource mapping can be expressed as:

$$\mathcal{D}_C(\delta_j) \xrightarrow{\mathcal{C}} C_i \rightarrow \mathcal{D}_{T,C_i}(\delta_j) \tag{3.1}$$

where $\mathcal{D}_C(\delta_j)$ denotes content descriptor of δ_j.

Resource mapping, based on content-based classification, can be summarized as follows. The content-based classification scheme $\mathcal{C}$ clusters activity periods based on a set of content features contained in content descriptor $\mathcal{D}_C$ into a set of content classes. Video traffic that is generated over the given activity period is modeled by the traffic model that is associated with the content class. In practice, it assumes that activity periods with the same content class are associated with the same traffic model. We will present experiments to verify this assumption later in this section.

The content-clustered model was investigated as proof of the proposed content-aware framework. To demonstrate the effectiveness of the model, the following experiment was conducted. In the experiment, manual methods of content extrac-

tion relying on subjective analysis and classification were used. The goal was to study performance achievable by the subjective content-based traffic model while isolating possible errors made by the automatic content analyzer. The content features were selected such that they can be used to visually describe the scene. Main applications of the content-clustered model is synthetic traffic generation based on scene description scripts.

The purpose of this experiment was not to select an optimal combination of parameters describing the video content, but to show that even this simple model can reveal the correlation between the video content and the trace. First, using manual frame-by-frame control of the MPEG-2 video player, we visually observed a video and segmented it into activity periods by detecting abrupt and substantial changes in the video content. Second, we characterized each activity period by a set of content features.

3.3.1 Estimation of Content Features

Table 3.1 summarizes the content features and their evaluation, as they were used in the content-clustered model. The content features were chosen such that they can easily be obtained by direct visual observation of the video. A single global feature, camera operation, was chosen to characterize the activity period in the global sense. Each activity period was manually classified as camera static, camera panning or camera zooming.

Within each activity period, a maximum of three video objects were identified. To limit the number of classes and control the complexity of the content-based classification scheme, both global and object features were categorized into three values only. Three object features have been used: object size, spatial complexity, and motion. Independent of location, each VO has been assigned an approximate

content features $\mathcal{F}_G$, $\mathcal{F}_O$	descriptor	category
camera operation	$\mathcal{D}_G$	static, panning, zoom
number of video objects	-	1, 2, 3
video object size	$\mathcal{D}_O$	small, medium, large
video object complexity	$\mathcal{D}_O$	smooth, medium, cluttered
video object speed	$\mathcal{D}_O$	low, medium, fast

Table 3.1: Estimation of global and object features.

size of 33%, 66%, and 100% of the entire image (i.e., frame size), corresponding to small, medium, and large object size. Video object motion* was also manually estimated and categorized into one of three categories: slow, medium, and fast. Spatial complexity was approximately categorized as smooth, medium, and cluttered.

3.3.2 Content Classification

Figure 3-12 depicts three content-based classification schemes suitable for MPEG-2 video. The model A incorporates all content features that were determined using the described manual procedures. It includes three different classifiers A1, A2, and A3 corresponding to one, two, and three objects in the activity period. Classifiers differ in the number of content features used for classification. For example, classifier A2 is selected when two video objects are identified in the activity period. In that case, the activity period is classified according to camera operation, motion and spatial complexity of each video object. The disadvantage of this classification scheme is the large number of classes ($3^3+3^5+3^7 = 2457$). It is therefore desirable to decrease the number of classes.

The number of classes can be simplified in the following way. First, the number of identified video objects was limited to two. Second, the spatial complexity of the

*Note that object motion here refers to perceived motion rather than "true motion". For example, in the activity period, where a camera is tracking a foreground object, perceived speed of the foreground object is close to zero.

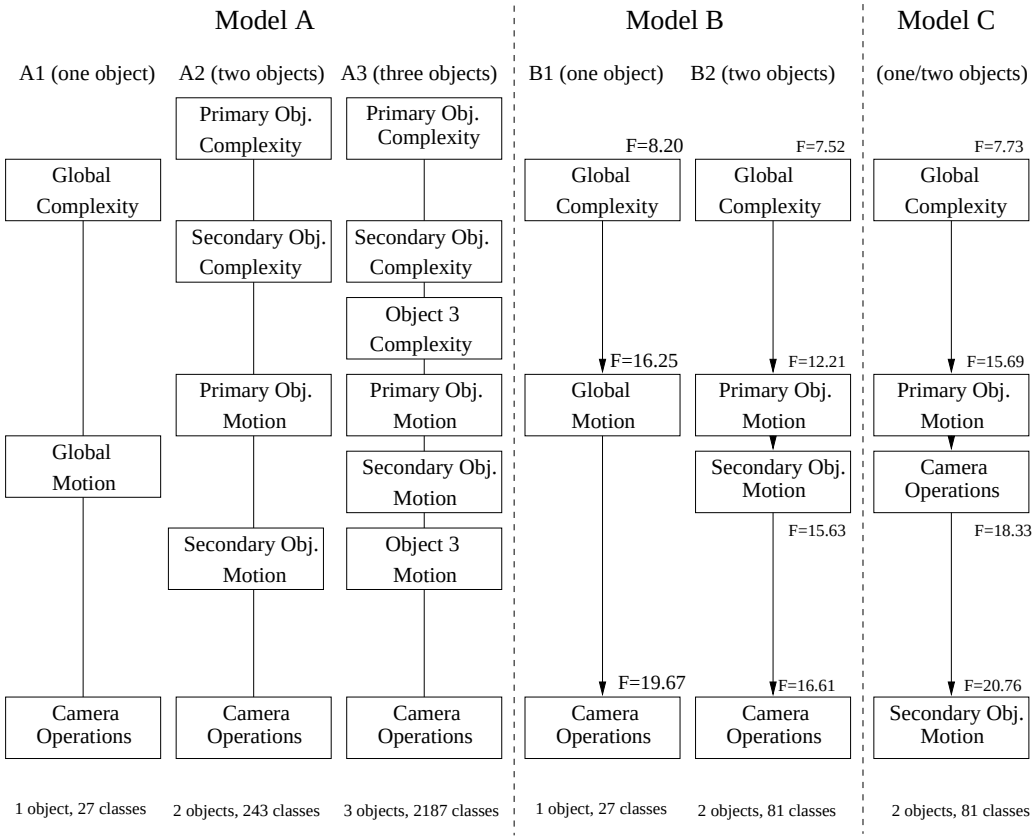

Figure 3-12: Manual activity period classification.

entire image instead of the object was used. The classifier is depicted as model B in Figure 3-12. This way, the number of classes was reduced to 108 (3^3+3^4). To further simplify the model, a final classifier, depicted as model C in Figure 3-12, was used for activity periods with single object (background only) and two objects (background and single video object). For the single object case, motions of composite objects (e.g., Primary and Secondary) were considered equal. The number of classes of model C was reduced to 81.

3.3.2.1 Classification Consistency Method

We have introduced a content-based classification scheme that clusters activity periods according to their content. To validate the relationship between video content and its bandwidth requirements, we propose the following method which measures the classification consustency. The classification consistency method was applied to the manual classification scheme, described in detail in Section 3.4.

The classification consistency method measures the "accuracy" of classification. In particular, it measures (i) the similarity of bandwidth requirements of activity periods identified as belonging to the same class and (ii) how distinct they are for activity periods classified into different activity classes. The classification consistency method can be used to measure the "goodness" of a classification scheme, e.g., how well activity periods of the similar content can be mapped into same traffic descriptor classes. Note that the proposed classification consistency method is not intended to measure clustering consistency in terms of content features. On the contrary, it is intended to measure clustering consistency in terms of traffic descriptors.

Assume that the bandwidth requirements of the activity periods (e.g. $\mathcal{D}_T$ descriptors) can be parametrized and written as vector $\vec{X}$. The vector $\vec{X}$ can also define a point X in a P-dimensional space $\mathcal{S}^P$. In general, we can define distance between vectors $\vec{X}$ and $\vec{Y}$ in the space $\mathcal{S}^P$ as the sum of weighted differences with power b: $\sum_{i=1}^{P} w_i |y_i - x_i|^b$. In the case where traffic descriptors are represented by D-BIND traffic descriptors, this distance actually represents difference in resource requirements that must be allocated to two different activity periods.

Further assume, that each point X, which represents a particular activity period, is characterized by content features. Denote $\mathcal{P} = \bigcup_{n=1}^{g} p_n$ as a set of g partitions, each representing a specific content class. Note that the partitions are mutually exclusive and it holds that $p_i \cap p_j = \phi$ for $i \neq j$. In other words, each point $X \in \mathcal{S}^P$

is classified and assigned into a single partition p.

Define *within class distance* $\mathcal{D}_{in}(p)$ and *between class distance* $\mathcal{D}_{out}(p)$ in the following way:

$$\mathcal{D}_{in}(p) = \frac{1}{2} \sum_{X_i, X_j \in p} \Delta^2(X_i, X_j) \qquad \mathcal{D}_{out}(p) = \sum_{X_i \in p, X_j \notin p} \Delta^2(X_i, X_j) \tag{3.2}$$

where Δ^2 is Euclidean distance between points in $\mathcal{S}^P$

Note that Equation 3.2 defines evaluation of $\mathcal{D}_{in}(p)$ and $\mathcal{D}_{out}(p)$ for general partition p. It does not suggest how the set of partitions $\mathcal{P}$ was in fact obtained. Using Equation 3.2, define two measures on the partition set $\mathcal{P}$, namely the *degree of grouping* $G(\mathcal{P})$ and *degree of separation* $S(\mathcal{P})$:

$$G\{\mathcal{P}\} \triangleq \frac{1}{\mathcal{D}_{in}(\mathcal{P})} \sum_{p_k \in \mathcal{P}} \mathcal{D}_{in}(p_k) \qquad S\{\mathcal{P}\} \triangleq \frac{1}{2\mathcal{D}_{in}(\mathcal{P})} \sum_{p_k \in \mathcal{P}} \mathcal{D}_{out}(p_k) \tag{3.3}$$

where $\mathcal{D}_{in}(\mathcal{P})$ is within class distance measured on an entire set of partitions $\mathcal{P} = \bigcup_{n=1}^{g} p_n$.

The degree of grouping conveys the information regarding how well points $X, Y \in \mathcal{S}^P$ are unified together under given partitions $\mathcal{P}$ while the degree of separation indicates how well partitions separate these points from each other. Ideally, the degree of grouping should be as small as possible and the degree of separation should be close to one. Note that $G\{\mathcal{P}\} + S\{\mathcal{P}\} = 1$.

The overall goodness of the content-based classification method can be obtained using the *classification consistency* $\mathcal{F}\{\mathcal{P}\}$, defined on a partition set $\mathcal{P}$:

$$\mathcal{F}\{\mathcal{P}\} \triangleq 10 \log_{10} \frac{S\{\mathcal{P}\}}{G\{\mathcal{P}\}} \tag{3.4}$$

Note that $\mathcal{F}\{\mathcal{P}\} \to \infty$ if each sample (i.e., activity period) is classified into a

separate activity class and $\mathcal{F}\{\mathcal{P}\} \rightarrow -\infty$ if no classification is performed (e.g., there is only one single class into which all samples are classified). Given the constraints on the number of activity classes, our objective is to maximize the classification consistency $\mathcal{F}\{\mathcal{P}\}$ of the proposed classification scheme.

The number and length of the intervals in D-BIND is one of the design parameters. Therefore, we assume they are fixed for computational convenience. In the case where number and size of the intervals do not match, appropriate transformation (interpolation) can be applied.

To find the relative importance of different content features in terms of their relation to bandwidth requirements, the classification consistency method was applied to the content-based classification scheme of model B and C. Starting from top to bottom, features are ordered in terms of their influence on the bit-rate. For example, to determine the content feature that most influences the resource requirements of the activity period, all three content features were compared: spatial complexity, motion, and camera operations. The spatial complexity was selected because it achieved the highest value of classification consistency, $\mathcal{F} = 7.73$. Figure 3-13 depicts the derivation of the optimal classification tree for model C. Based on this method, the order of importance of content features was identified as follows: spatial complexity, motion and camera operations.

The model C has 81 classes only. Despite a reduced number of classes, in this example, the classification tree of model C achieved better classification consistency when compared with Model B. The model C classifier was used in network simulations, described in the Chapter 4.

The order of importance has significant implications for the design of a real-time content analyzer/classifier. For example, because of the MPEG-2 motion prediction, spatial complexity can be directly evaluated in the compressed domain for I frames

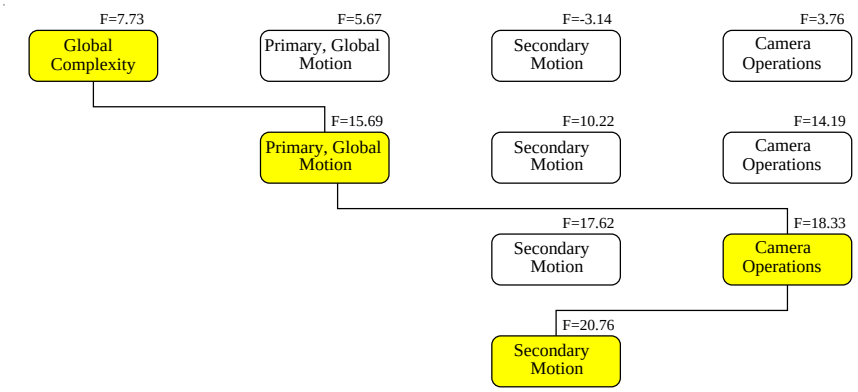

Figure 3-13: Experimental MPEG-2 content-based scene classification tree (Combined First and Second layer, 1-2 objects).

only. However, since spatial complexity is important in terms of its influence on the bit-rate, it is also desirable to predict its value for frames for which direct evaluation is not possible (i.e., P and B frames). Simple prediction can be realized using a group of frames (GoP) for which spatial complexity is assumed to be constant during the entire GoP period. This prediction may be satisfactory in the middle of the activity period. However, in cases when a new activity period starts in the middle of the GoP, full frame decoding may be necessary to estimate spatial complexity for beginning frames of the new activity period.

3.3.3 Video Segmentation

Video segmentation algorithms have been previously applied for detection of scene changes in content-based video retrieval systems. In addition, video segmentation algorithms have been used for determining re-negotiation points in dynamic resource allocation.

Generally, we distinguish two video segmentation algorithms: off-line and on-line [53, 54, 13]. The off-line video segmentation algorithm can be applied to stored video streams. Since off-line algorithms are not causal and can take advantage of

knowledge of the full video trace history, off-line segmentation algorithms can be used to obtain the optimal re-negotiation schedule. However, since determination of the optimal re-negotiation points is computationally extensive, sometimes heuristic off-line segmentation algorithms are used [13]. These algorithms are typically less computationally extensive but lead to sub-optimal solutions only.

On-line video segmentation algorithms used for real-time traffic are causal; they are based on an assumption of knowledge of only present and previous bit-rate history. These algorithms are based on heuristic traffic prediction models monitoring the incoming traffic, network queue length or cell loss to assess the future stream resource requirements.

Review of previous video segmentation algorithms has shown that their criterion for segmentation was based solely on a traffic profile. It is rather difficult to find parameters of traditional stochastic models and to find the singularity points, e.g., points at which characteristics of the traffic change. To solve this problem, various heuristics were used to determine these points. However, another approach can be taken. Because of the close relation between video content and video traffic, video content can be used to determine the periods at which traffic characteristics change. The content-based video segmentation algorithm is based directly on video content rather than bit-rate.

A video stream is segmented in the following way. First, each frame is segmented into video objects. Global features and object features are estimated. Then, each frame is classified according to content features. The consecutive frames being classified into the same content class are grouped together to form a single activity period. A scene change occurs between two frames classified into the different classes.

Formally, the content-based segmentation is expressed as follows. Denote video $V = \{f_k\}_{k=1}^{n}$ as a sequence of frames f_k. Denote C_i content class of activity period

δ_j. Denote content-based classification of the frame as $f_k \rightarrow C_i$ and classification of the activity period as $\mathcal{D}_C(\delta_j) \xrightarrow{\mathcal{C}} C_i$. The goal of the content-based segmentation is to partition a video into a set of non-overlapping segments (activity periods) $\delta j = [f_k, f_{k+l_i}]_j$ of length l_j. Each segment consists of a variable number of frames. We say that an activity period is classified into the class C_i when the following holds:

$$(\mathcal{D}_C(\delta_j) \xrightarrow{\mathcal{C}} C_i) \quad \equiv \quad \{\forall f_k \in \delta_j, \quad f_k \rightarrow C_i\}, \qquad C_i \in \mathcal{C} \tag{3.5}$$

where $\mathcal{C}$ is the set of all different content classes. In other words, content-based segmentation divides the video frame sequence into segments, frames of which belong to the same scene class.

3.3.3.1 Detection of Activity Periods for MPEG-2 streams

Detection of activity periods was based on a simplified algorithm for scene change detection in MPEG video streams [59, 65]. This algorithm is based only on partial decoding of the compressed video stream. Since the full decoding of each frame is not necessary, the computing complexity can be reduced. In particular, a simple MPEG-2 video parser that extracts DCT DC coefficients for I- and P-frames and motion vectors for P- and B-frames will suffice for its implementation.

The algorithm for detection of activity periods depends on the frame format. The algorithm for scene detection in I-frames is based on the fact that intensity variance of frames within the same activity period tends to be stable. Transition of activity periods is indicated by a peak of the absolute value of the frame difference. Additionally, the algorithm in [65] suggested as a second condition to use ratio of a number of forward to backward motion vectors of proceeding B-frames to identify abrupt "scene" changes in I-frames. The reason for this additional condition was

that the absolute value of frame difference may be unstable at periods of high motion.

The activity period detection algorithm for P-frames is based on the ratio of the number of macroblocks without motion compensation (i.e., intra-coded macroblocks) to the number of blocks with motion compensation. The reason for this measure is that when the activity period change occurs at P-frame, many macroblocks must be intra-coded because motion compensation cannot find corresponding macroblocks at previous anchor frame. Activity period change is declared at peaks of this ratio.

Detection of activity periods at B-frames is based on the ratio of the number of backward to forward motion vectors. When activity period change occurs at B-frame, most motion vectors come from the future anchor frame (e.g., later in display order) rather than from the past anchor frame. The adaptive window-based threshold technique is used to detect peaks. Activity period change is declared at peaks of this ratio.

3.3.4 Resource Mapping

To illustrate results of the content-based scene classification scheme in terms of class-to-resource mapping, D-BIND traffic functions corresponding to different classes have been evaluated. In particular, three D-BIND traffic functions together that were obtained using three quantization levels of complexity (i.e., smooth, medium, and cluttered), are depicted in Figure 3-14. Error bars indicate standard deviation of D-BIND coefficients at each time-scale. Resource descriptors obtained with the use of two content features (complexity and motion) are depicted in Figure 3-15. A, B, and C are major groupings relating to spatial complexity and "slow/medium/fast" are minor groupings relating to motion.

Figure 3-16 depicts an increase in classification consistency with the number

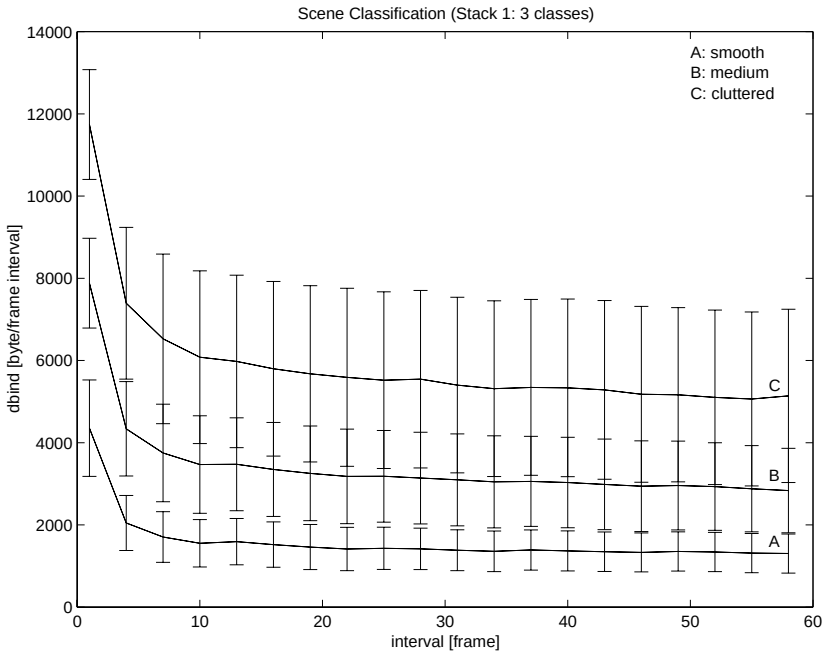

Figure 3-14: D-BIND traffic resource mapping based on three complexity classes.

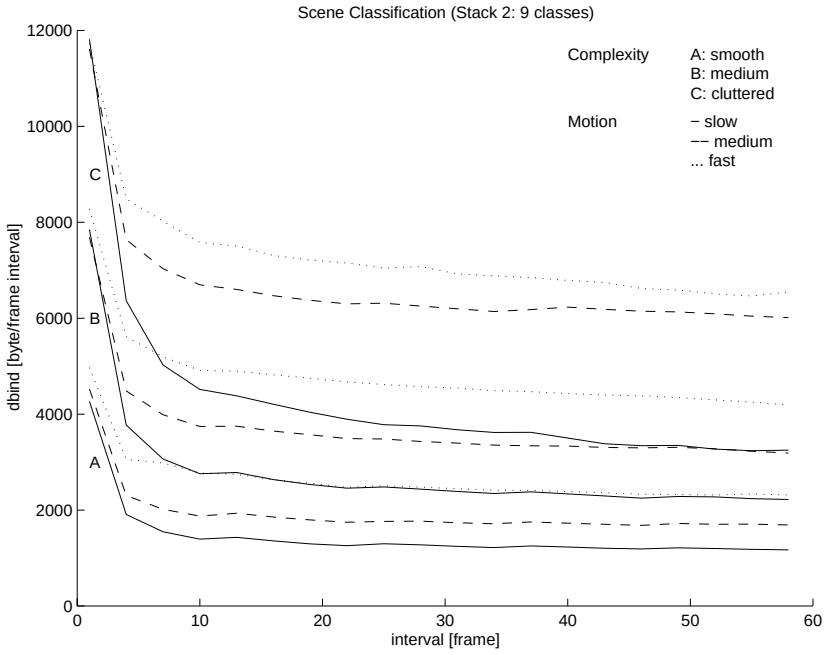

Figure 3-15: D-BIND traffic resource mapping based on complexity and global motion.

of categories of spatial complexity. Classification consistency is not substantially improved beyond nine categories, reaching $\mathcal{F} = 12.22$. At this point, further improvement is possible only if the number of content features is increased.

Figure 3-17 shows an increase in classification consistency with an increase in the number of content features. Four content features were used in the following sequence: global complexity, primary object motion, camera operations, and secondary object motion. All content features were classified into three categories. For example, using two content features classified into three categories each, classification consistency increases to $\mathcal{F} = 15.69$. Note that this value is higher than the case in which only a single feature was used ($\mathcal{F} = 12.22$ refer to Figure 3-16). The use of two different content features, classified into a small number of categories, is advantageous as compared to the use of a single content feature, classified into a large number of categories.

Previous results suggest that while an increase in the number of categories of each individual content feature improves the classification consistency, such improvement is not linear and asymptotically approaches a limiting value. The use of more content features, categorized into fewer categories each, has shown to be more effective when compared to classification schemes based on fewer content features, classified into a larger number of categories each.

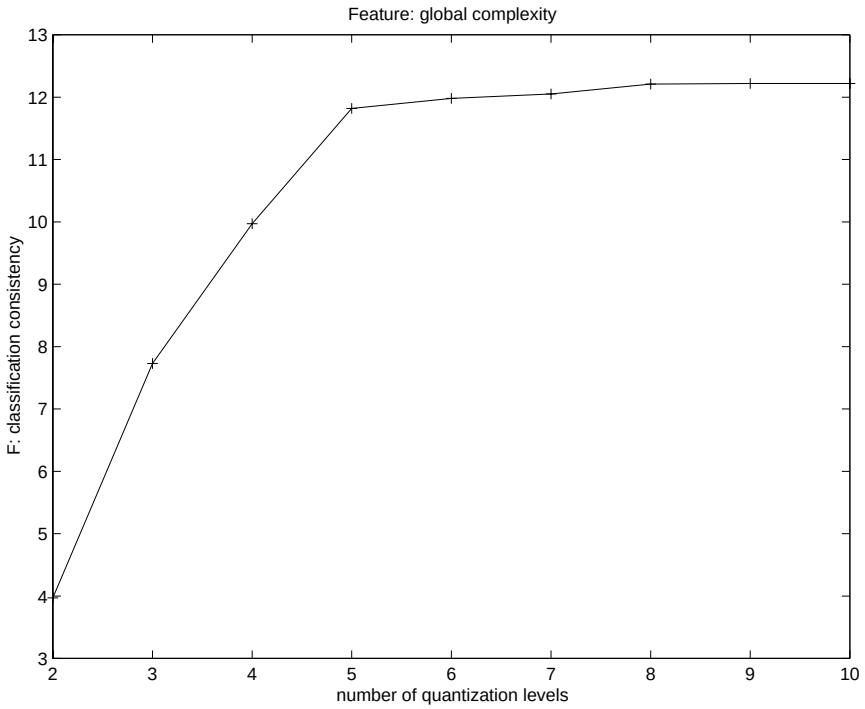

Figure 3-16: Relation between the number of feature quantization levels and $\mathcal{F}$.

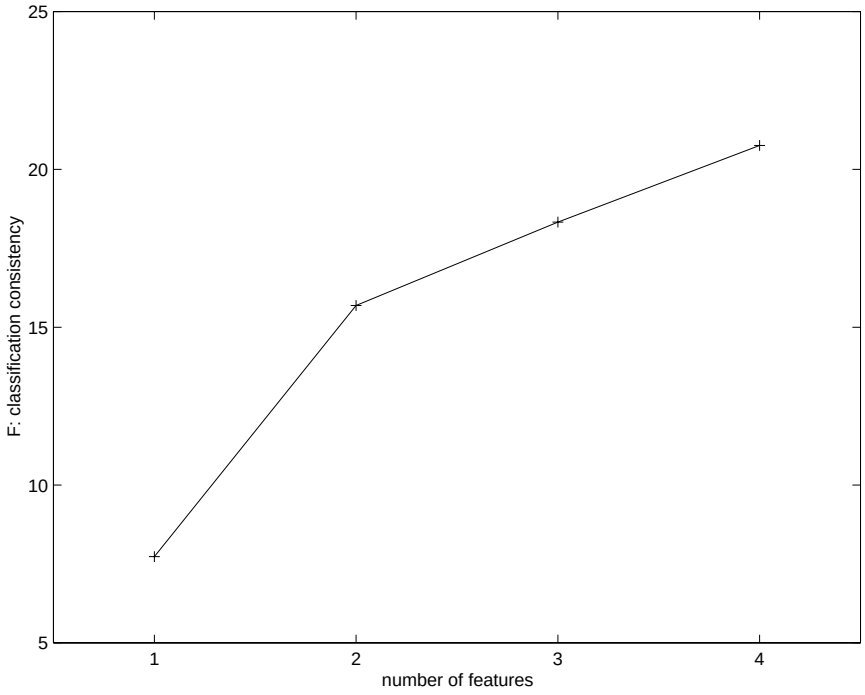

Figure 3-17: Relation between the number of content features and $\mathcal{F}$.

3.4 CB Resource-Clustered Model (CBRC)

We have found that although the content-clustered model, described in Section 3.3.2, is sufficient for manual feature estimation, it does not perform well in a case where features are extracted automatically. Note that in the content-clustered model, activity periods were classified into content classes based on content features.

The resource-clustered model assumes that activity periods are classified into traffic classes T_i. Denote $\mathcal{T} = \{T_i \mid i = 1, 2, \cdots, l\}$ a content-based classification scheme consisting of l traffic classes T_i. Denote $\mathcal{D}_{T,T_i}(\delta_j)$ a characteristics traffic descriptor of activity period δ_j that is associated with each traffic class T_i. Then, the resource-clustered scheme can be expressed as:

$$\mathcal{D}_C(\delta_j) \xrightarrow{\mathcal{T}} T_i \rightarrow \mathcal{D}_{T,T_i}(\delta_j) \tag{3.6}$$

where $\mathcal{D}_C(\delta_j)$ denotes content descriptor of δ_j. In other words, traffic classes are directly predicted based on content features. Content classes are not formed.

3.4.1 Traffic Classification

The resource-clustered classification scheme utilizes machine learning algorithms. It operates with two modes: training and selection. In the training mode, the content-based classification scheme is initialized: the decision tree classifier and characteristic traffic descriptors, $\mathcal{D}_{T,T_i}$, are estimated. In real-time, the scheme operates at the selection mode: the activity period's traffic class is determined by using the decision tree classifier. Corresponding $\mathcal{D}_{T,T_i}$ is used to predict bandwidth requirements for that activity period. For flexibility, in our experiments we have used general, publicly available machine learning tools. We have simulated generation of the decision tree classifier offline using a subset of activity periods only. The remaining activity

periods were used for testing.

The decision tree classifier was generated in the following way. First, the activity periods were automatically clustered into different traffic classes T_i by Autoclass III [78]. Autoclass III is a Bayesian unsupervised classifier that estimates class membership given unlabeled test cases. Each activity period was classified into one of nine traffic classes.

Figure 3-18 illustrates the classification results. In this example, 9 classes were formed based on traffic descriptors of 295 activity periods. Each sub-graph illustrates individual traffic descriptors within one single class, which are shown in the shaded curves. The single dark curve represents the characteristic traffic descriptor of that class, which is obtained, in this case, as the 90 percentile of the traffic descriptors belonging to that class.

The classes are numbered 0 to 8. The number in parenthesis indicates the number of traffic descriptors in that class. In our case, the number of traffic descriptors in a single class ranges from 28 to 47. The figure clearly shows the good clustering performance of Autoclass III software, as the traffic descriptors of similar shapes are clustered into the same traffic class.

After the clustering operation, traffic classes and content features were used to generate the decision tree classifier. The decision tree classifier was estimated offline by OC1 software [79], a supervised machine learning system based on oblique decision trees. Decision trees of this form consist of a linear combination of the attributes (in our case content features) at each internal node and can be viewed simply as a more general form of axis-parallel univariate decision trees. In our experiments, a relatively high classification accuracy of 86.1 % was achieved. Accuracy of classification for each traffic class is summarized in Table 3.2.

Figure Figure 3-19 illustrates the classification results based on the decision tree.

class	1	2	3	4	5	6	7	8	9
accuracy %	85.1	88.9	82.9	87.5	83.3	90.00	89.7	82.1	85.7

Table 3.2: Accuracy of decision tree classifier.

The sub-graphs are presented in the same way as the ones in Figure 3-18.

The decision tree enables estimation of the traffic class given a set of content features that are easy to obtain from compressed video streams. However, a mismatch between content features and traffic class can occur. In other words, in some instances, a decision tree is not able to correctly identify a traffic class based on content features alone. This effect can be observed from Figure 3-19. Figure 3-19 depicts traffic descriptors that were not correctly classified. In practice, this will lead to choosing a wrong characteristic traffic descriptor. The first number in parenthesis indicates the number of traffic descriptors that were classified into that class; the second number indicates the number of traffic descriptors among then that were classified incorrectly.

The decision tree was used in our simulations to predict resource requirements for each activity period. In particular, once the traffic class of an activity period was determined by the decision tree classifier, the characteristic traffic descriptor $\mathcal{D}_{T,T_i}$ was used to predict the resource requirements for that activity period. In our simulations, we have used three different metrics for an evaluation of $\mathcal{D}_{T,T_i}$: mean, maximum and 90 percentile of a particular class.

The resource clustered model can be used for the prediction of bandwidth requirements in a dynamic resource allocation system. The approach and its performance is further discussed in Chapter 4. In this model, we have used a real-time content analyzer that is based on fully automated methods of content analysis. It was intended to provide practical experience with the implementation of an automatic content analyzer/classifier.

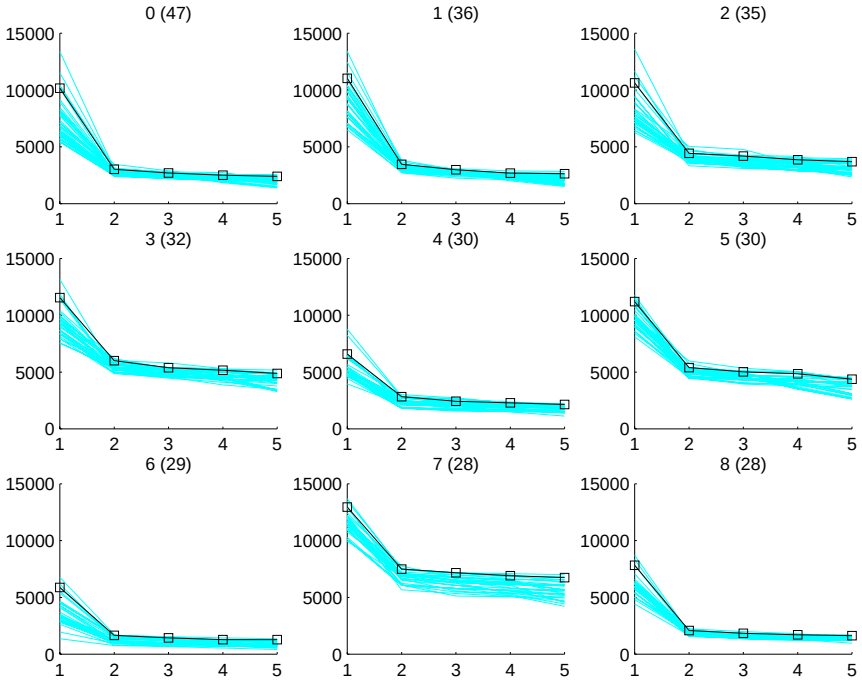

Figure 3-18: Traffic classes.

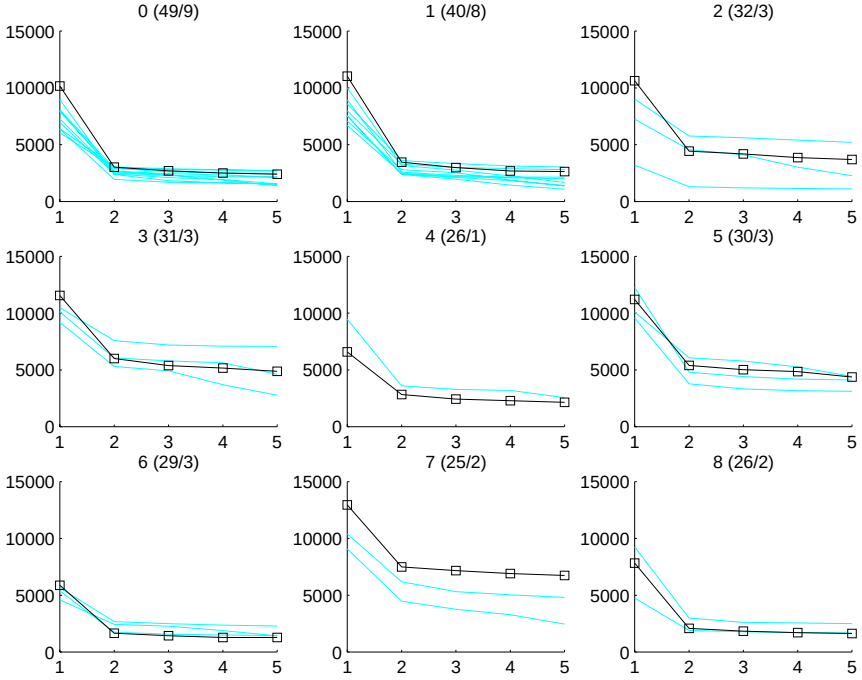

Figure 3-19: Accuracy of content-based decision tree classifier.

Chapter 4

Network Resource Allocation

4.1 Introduction

Historically, telephone circuit switched networks provided guaranteed service while packet networks provided best effort service. Both networks were not designed to support quality of service (QoS) and multimedia in general. Use of circuit switched networks for multimedia is not economical, mainly because quality of service requirements and highly variable traffic characteristics would lead to low network utilization. On the other hand, packet networks are more effective in terms of use of network resources. However, to support multimedia efficiently, quality of service provisioning is necessary.

A comprehensive survey of QoS architectures can be found in the literature [68]. In general, various classes of quality of service can be designated for transmission of a video, audio and data over multimedia networks. For example, with respect to user QoS requirements, application QoS requirements can be loosely divided into the following categories: hard delay and loss bounds (e.g., CBR, VBR video), hard delay and soft loss bounds (e.g., voice), soft delay and hard loss bounds (e.g., real-time data), and soft delay and loss bounds (e.g., data) [69]. While network QoS is commonly specified in terms of cell loss, end-to-end delay, and cell delay

variation (CDV), application QoS requirements cannot be expressed in those terms. Mapping between the application and network QoS requirements is necessary. The mapping may not be unique; application QoS requirements may possibly be satisfied by different network QoS descriptors.

The role of the call admission control (CAC) is instrumental in QoS provisioning [69, 27, 28]. CAC decides whether to admit or reject a new session and manages network resources such that QoS is guaranteed for all accepted sessions. The assignment of available network bandwidth and buffers among sessions is, in general, policed at the network interface such that no stream can corrupt or cause QoS violation of other well-behaved streams. On the session level, the call admission is based on network parameters (e.g., network status and desired network utilization) and parameters supplied by an individual session such as service class, traffic model (descriptor), QoS objectives, etc. For example, for ATM traffic, five service classes are defined by ATM Forum: constant-bit rate (CBR), real-time variable-bit rate (rt-VBR), non real-time variable-bit rate (nrt-VBR), available-bit rate (ABR), and unspecified-bit rate (UBR) [14].

Because of constant picture quality, support of VBR video in packet networks is desirable. However, because of complex VBR traffic characteristics, call admission control for VBR service class is a challenging issue. Recent studies show that transport of VBR video may lead either to network congestion or low network utilization [15]. These results highlight the necessity of new advanced resource allocation techniques suitable for both real-time and non real-time VBR video traffic.

One promising solution, aimed to overcome these difficulties, is to replace the static resource allocation of CAC, with dynamic resource allocation (DRA). In static resource allocation, the network resources are allocated only once at the beginning of the session. Resource allocation is done as part of CAC algorithm. On the contrary,

DRA allows in-call resource allocation. The network resources, based on current need, can be dynamically requested while the session is in progress. Compared to the static resource allocation, DRA may lead to an increase in network utilization.

In essence, DRA is a scheme for preventive congestion control that supports quality of service (QoS) for individual sources. DRA is suited for transport of VBR video over bandwidth limited networks, e.g., for networks that can support only a limited number of streams (in the order of several tenths) for which statistical multiplexing would not lead to substantial multiplexing gain. It is efficient for bursty, non-homogeneous VBR sources for which the traffic model is generally difficult to obtain. For example, DRA can be beneficial in access and wireless networks with limited or variable bandwidth that can support a relatively small number of concurrent VBR video streams.

One of the challenges of dynamic resource allocation exists in the selection of renegotiation strategies. The solution is connected to traffic prediction and selection of renegotiation points, i.e., instances in which renegotiation of resources should take place. The selection of renegotiation points is more complicated for real-time video. In the past, several traffic prediction models based on a single indicator (bit rate) have shown promising results [52, 53, 54]. However, since these models are based on heuristic traffic prediction methods combining traffic monitoring and buffer occupancy, their performance is sensitive to parameter selection.

In this chapter, we present another approach. In particular, we apply the content-based framework to both traffic prediction and selection of renegotiation intervals. We refer to this new approach as content-based DRA. The content-based traffic model explores the video content of the video as an important indicator of VBR stream bandwidth requirements. Because video content can be extracted from video streams in real-time, traffic prediction based on video content is suitable for

live video.

The rest of this chapter is organized as follows. In Section 4.2, we discuss more specific issues related to the congestion control and resource allocation. In particular, we point out the need for the layered congestion control matching the different traffic patterns in both core and external networks. We present an overview of different traffic models that are typically used in the admission and congestion control algorithms. In Section 4.3, we present our content-based dynamic resource allocation model, its real time implementation and related issues of video segmentation, classification, and resource mapping. We also compare results of trace-driven simulation of content-based model to results obtained by using other models.

4.2 Resource Allocation

The function of the call admission control is to avoid congestion in communication networks. When the user requests a new network connection, the admission control decides if a new call can be accepted and a connection established. A new call is accepted only if there are enough resources and the QoS of all multiplexed connections will not be violated. In addition to network resources, other factors might be considered in call admission. For example, only calls maximizing the network utilization or network provider's profit would be admitted.

For video transport over packets networks to be effective, the following issues should be addressed. First, high burstiness and the multiple-time scale property of VBR traffic prevents achievement of high statistical multiplexing gain when the number of multiplexed streams is small [55]. This applies mostly to bandwidth-limited networks that can support only a small number of streams. Second, because bursts of high bit rate in VBR streams can occur for a relatively long time, large network buffers associated with large delay would have to be used to allow transmis-

sion at lower than peak rate. Under these conditions, the traditional static resource allocation is inefficient (requires near peak rate bandwidth allocation) leading to very low network utilization. To keep the delay small and still provide adequate QoS, controlled access to the network resources on the burst time-scale is necessary.

For access networks or network interfaces that cannot accommodate large number of video streams, a more flexible multilayer congestion control model for transport of bursty traffic was proposed [51]. In this model, the congestion is controlled at three layers: packet (cell), burst and call. Congestion at each layer is measured by probability of cell loss, burst blocking and call blocking probabilities. Traffic control at the higher layer can reduce congestion at the lower layer. At the call layer, a new session is denied if the excessive burst blocking probability occurs; similarly, the burst of the already admitted call is denied if it would cause excessive cell loss at the cell layer. For example, the network may support probabilities in the range of 10^{-2} for call blocking, 10^{-2} to 10^{-4} for burst blocking and 10^{-6} to 10^{-8} for cell loss.

4.2.1 Dynamic Resource Allocation

Generally, we can identify two resource allocation mechanisms: static and dynamic. Static allocation uses a priori traffic source descriptor and is concerned with resource allocation on the call layer. During the connection setup, the network resources are allocated at the beginning of the call and deallocated only at the end of the call.

On the other hand, dynamic resource allocation is based on the three-layer congestion control model. It is a technique allowing network resources to be allocated on need basis during the lifetime of the connection. A request for the change of resources is generated when source traffic conditions change; for example, at the beginning of excessive cell bursts. While a request for a decrease in resources will

always be granted, the request for an increase can be denied.

In theory, dynamic resource allocation represents a burst resource allocation layer [51]. This layer operates on the time-scale between cell scheduling and call admission layers.

The efficiency of dynamic resource allocation depends on the strategy of determining the renegotiation intervals and accuracy of traffic prediction. The most efficient renegotiation scheme is to allow renegotiation in every time-frame interval [52]. However, there is an overhead associated with the renegotiation, namely the processing cost of a renegotiation request message. The implementation of a frame-by-frame renegotiation scheme could create a bottleneck due to the limited message processing power at the switch or router.

Figure 4-1 depicts the conceptual model of a dynamic resource allocation system. Its two primary components are the resource allocation broker (RAB) and media traffic agent (MTA). RAB controls bandwidth allocation for multiple streams that share network resources. MTAs are associated with video streams. MTA predicts video stream bandwidth requirements, generates resource requests, and, if necessary, dynamically regulates the video stream such that it conforms to its currently assigned bandwidth. The location of RAB and MTA components in the network is flexible. For example, both components can coexist at a user network interface UNI. In that case, dynamic resource allocation is performed locally at the network multiplexer only. Alternatively, RABs can be distributed across the network. They can be located at the external ATM switch, QoS enabled router or a wireless base station. In the distributed case, RABs control resource allocation locally along the route for each individual network link and node. In the centralized case, the single RAB controls the resource allocation among multiple switches.

The network resource and QoS management at the RAB is based on (i) call

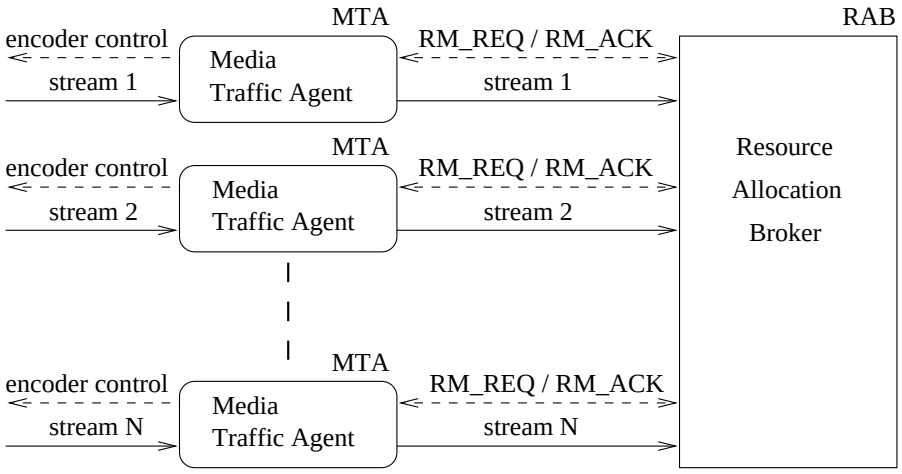

Figure 4-1: Conceptual model of DRA system.

admission control (CAC) and (ii) dynamic resource allocation (DRA) algorithms. The CAC algorithm handles the admission of new streams into the network [69]. Once the stream is admitted, its resource allocation is handled by DRA algorithm. For each admitted source, RAB maintains information about the desired QoS requirements and current network resource assignments.

DRA algorithm assigns the bandwidth for each individual video stream based on its requirements. In the case of stored video, future bandwidth requirements may be known in advance. However, in real-time video, the future resource requirements can only be predicted. In that case, the MTA assumes the role of traffic predictor. If the future resource requirements of the stream exceed the resources currently available, the MTA generates resource management requests, RM_REQ, and send them to RAB for processing. RM_REQs convey information about network resources that are requested by the video stream. In RAB, the DRA algorithm compares resource requirements with currently available network resources. If there are enough resources available, they are allocated (i.e., increased or decreased) and reserved. In that case, successful reservation is acknowledged by RM_ACK. Otherwise, request

is rejected and a negative response RM_NACK may be returned. To limit signaling overhead, some lightweight DRA protocols do not explicitly confirm or reject the reservation [49].

In the when that resource allocation brokers are distributed across the network, RM_REQs messages may impose signaling overhead on the network. In addition, the processing of requests at the RAB should also be considered. The frequency of renegotiation (e.g., frequency of generation of RM_REQ) plays an important role in the evaluation of the performance of different dynamic resource allocation schemes. Frequency of renegotiation in the order of seconds (i.e., interval between renegotiations) is reasonable and possible to achieve with current technology.

To lower signaling and procession overhead, new lightweight DRA reservation protocols were recently proposed; for example, YESSIR reservation mechanism that is suitable for Internet. It is built on top of RTP and RTCP protocols and supports in-band DRA reservations [48]. Another reservation mechanism, SRP, allows applications to make dynamic reservations of their minimum bandwidth using two protocols [49]. The reservation protocol uses data packets of different types. Packet type is used to convey reservation information from the source to the network. The feedback protocol conveys information about reservation status back to the sender.

4.2.2 Video Segmentation and Resource Prediction

Dynamic resource allocation systems may employ different video segmentation algorithms, traffic prediction models, and signaling protocols. In addition, different traffic descriptors may be used for resource allocation. The video segmentation algorithm determines renegotiation points, i.e. times at which renegotiation should take place. The traffic prediction model determines resource requirements that should be requested. The traffic prediction is needed for on-line segmentation only; in off-

line segmentation, precise bandwidth requirements can be pre-computed and stored along with the video stream.

The renegotiation intervals are found by video segmentation algorithms that divide video streams into variable size sections [53, 54, 13]. Renegotiation points for stored video can be found off-line [44]. However, off-line segmentation algorithms cannot be directly used for real-time or interactive video for which future resource requirements cannot be precisely predicted. For real-time video, heuristic on-line segmentation algorithms are used instead [25]. On-line segmentation algorithms are causal; typically, they are based on heuristic prediction models monitoring the incoming traffic, network queue length or cell loss to assess the future resource requirements. In essence, their function is as follows: when current or future resource requirements exceed the reserved resources, more resources are requested. On the other hand, when the stream resource requirement is less than currently reserved, the request to decrease the resources may be generated.

The autoregressive traffic models are used for the real-time bandwidth prediction. Parameters of the traffic model are estimated in real-time. To improve the prediction accuracy, some heuristic models also use the status of the encoder or network buffer. The following are two examples of models used for traffic prediction.

Simple traffic prediction models can be realized as follows. Let X_n be a stochastic process and $\hat{X}_n$ be prediction of the X_n. The simple linear bit rate predictor can realized as follows:

$$\hat{X}_n = X_{n-1} + \sigma(X_{n-1}) \tag{4.1}$$

where $\sigma(X_n)$ denote a standard deviation of X_n. This prediction model assume an increase equal to the standard deviation. This may often lead to an overestimation of the bit rate.

A heuristic taffic prediction algorithm was suggested by Grossglauser [53]. It

uses a monitoring of video traffic and multiplexer queue length [53]. It is based on AR(1) model with additional term estimating required bandwidth to flush a current content of network buffer:

$$\hat{X}_n = \delta \hat{X}_{n-1} + (1-\delta)[X_{n-1} + \max\{b_i - B_h, 0\}] + \varepsilon \tag{4.2}$$

where ε is a white noise process, δ is an autoregressive coefficient, b_i and B_h are current buffer state and high buffer mark threshold; additional term $\max\{b_i - B_h, 0\}$ prevents buffer buildups.

Linear prediction algorithms work well only for processes for which the stochastic model is known and parameters can be easily obtained. Unfortunately, this is not the case in the VBR traffic modeling, where model parameters are constantly changing.

In the following, we discuss three DRA schemes: frame-based, renegotiated CBR (RCBR), and renegotiated VBR (RVBR). Both RCBR and RVBR schemes use different video segmentation and traffic prediction algorithms. We describe our content-based DRA in the next section.

4.2.2.1 Frame-Based DRA

In frame-based DRA, network resources (equivalent to number of bits in a frame) are requested in a frame by frame fashion. The main advantage of frame-based DRA is its ability to achieve high network utilization due to its high frequency of renegotiation. In addition, the size of the frame may be known in advance or it can be predicted relatively accurately. In general, network utilization depends on characteristics of the source as well as on the frequency of renegotiation; high-bursty VBR sources can benefit from an increase in renegotiation frequency while CBR sources cannot. The disadvantage of frame-based DRA is the high frequency

of renegotiation. The frame-based DRA scheme may not be practical if it causes bottleneck due to the limited processing capacity or bandwidth. [52].

High frequency of renegotiation can be compensated by simplified processing at the RAB. For example, note that frame-based DRA does not need complex VBR traffic descriptors to characterize its traffic at each period. In addition, lightweight resource allocation protocols can further reduce signaling overhead. For example, ATM block transfer that is being standardized by ITU-T allows bandwidth renegotiations on basis of a block of cells [47]. In transmission, the block of cells is enclosed in two resource management (RM) cells. The first RM cell requests network bandwidth and the second one releases bandwidth. At the network node, blocks of cells are handled as basic units, e.g., they are either accepted for transmission or fully discarded. Admission control and QoS support for frame-based DRA suitable for interactive video is presented by Lam et. al. [45] and Xie et. al. [46].

4.2.2.2 Renegotiated VBR Model

The renegotiated VBR (RVBR) dynamic resource allocation scheme, proposed by Knightly et. al. [56], is depicted in Figure 4-2. In RVBR scheme, VBR streams are characterized by D-BIND traffic descriptors. The heuristic on-line video segmentation algorithm of the RVBR scheme is based on the video traffic history. The current resource requirements of the stream are estimated on-line using a measurement-based algorithm of predetermined window length. If current resource requirements exceed the available bandwidth, resource management requests are generated immediately. Requests for reduction of resources are generated when currently estimated traffic falls under the given threshold.

In RVBR, the DRA algorithm used for accepting/rejecting of the resource allocation request is closely related to D-BIND video traffic descriptors. As described in

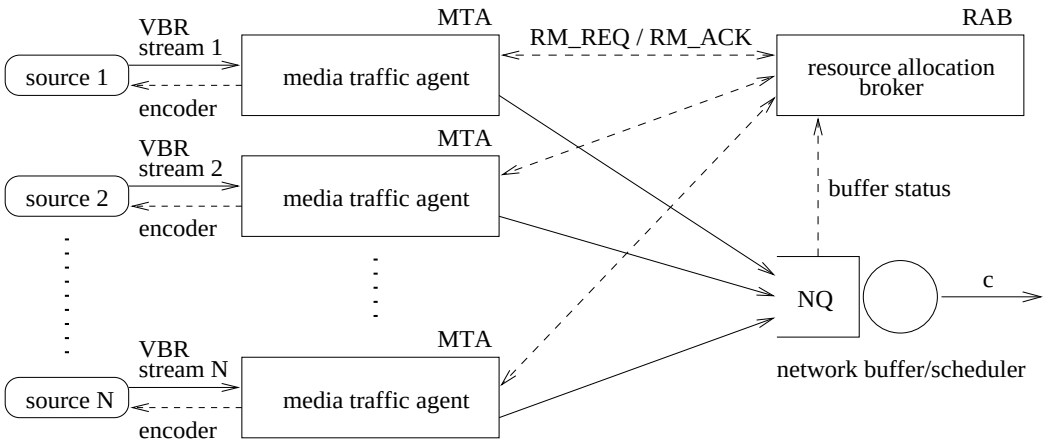

Figure 4-2: Content-based and RVBR dynamic resource allocation system model.

Chapter 2, D-BIND constrained sources s_i are characterized in terms of the traffic constrained function defined in terms of rate-interval pairs $R_T^{(i)} = \{(r_k^{(i)}, t_k) \mid k = 1, 2, \cdots, P\}$ [54]. Denote Q network buffer size and $B\tau$ network buffer requirements at time τ when new resource reservation request for the source s_{n+1} arrives. Further define subset $\mathcal{A}_\tau = \{s_i \mid i = 1, \cdots, n\}$ of n sources s_i that are currently enabled for transmission by dynamic resource allocation algorithm at time τ. For FCFS scheduling policy, the required network buffer size B_τ for all streams, including new one, is:

$$B_\tau = \max\{0, \max_k\{t_k(\sum_{\substack{i=1 \\ s_i \in \mathcal{A}_\tau}}^{N} r_k^{(i)} + r_k^{(n+1)} - c)\}\}, \qquad k = 1, \cdots, P \tag{4.3}$$

where N is total number of sources, admitted by CAC, and c is a link speed. Note that $n \leq N$, i.e., number of sources currently enabled by DRA is less or equal to total number of sources accepted by CAC. It is assumed in Equation 4.3 that $R_T^{(i)}$ of all sources are defined using same time intervals t_k.

The DBA algorithm for the RVBR system, used by a resource allocation broker, takes into account the current occupancy of a network buffer. It is formulated as

follows: denote Q_τ buffer occupancy at time τ. Any resource reservation request can be positively acknowledged only if currently available buffer space is greater than or equal to total buffering requirements of all sources B_τ:

$$B_\tau \leq Q - Q_\tau \tag{4.4}$$

4.2.2.3 Renegotiated CBR Model

The RCBR dynamic resource allocation scheme, as proposed by Keshav et. al. [53], is depicted in Figure 4-3. Contrary to other DRA schemes, in the RCBR scheme, streams are assumed to be separately buffered before entering a network multiplexer. Additionally, the network multiplexer does not use the shared network buffer. Video streams are multiplexed at the entrance to the network using the FCFS network scheduler policy. The RCBR scheme is essentially a piece-wise CBR allocation scheme. The DRA algorithm for RCBR is simplified, compared to RVBR, because the bandwidth descriptor is represented by a single value: CBR rate. If the sum of CBR rates of all already accepted streams and new ones exceeds the link capacity, the request is rejected. Otherwise, the request is accepted.

The RCBR on-line segmentation is closely related to the traffic model. It is assumed that video traffic between renegotiations can be described by an autoregressive AR(1) stochastic model:

$$\hat{X}_n = \delta \hat{X}_{n-1} + \varepsilon \tag{4.5}$$

where X_n denotes current stream bandwidth, $\hat{X}_n$ is the predicted stream bandwidth, ε is a white noise process, and δ is an autoregressive coefficient.

However, in RCBR the bandwidth is predicted using the following heuristics: the model uses an additional term that describes the current occupancy of network

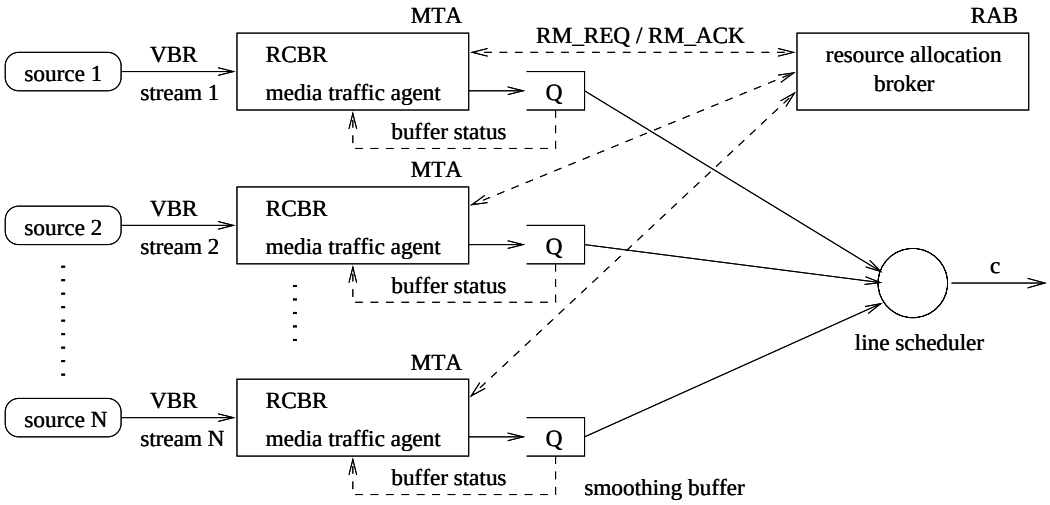

Figure 4-3: Simulation model for RCBR dynamic resource allocation.

buffer:

$$\hat{X}_n = \delta \hat{X}_{n-1} + (1-\delta)[X_{n-1} + \max\{b_i - B_h, 0\}] \tag{4.6}$$

where b_i and B_h are the current buffer state and high buffer mark threshold (in our experiments set to 90% of the buffer occupancy); the additional term $\max\{b_i - B_h, 0\}$ prevents buffer buildups.

4.3 Content-based DRA

In the content based DRA, resources are allocated based on the video content of activity periods rather than on heuristic traffic measurements. The bandwidth of activity periods is predicted by the content-based video traffic model. The content information can be extracted from the compressed video stream in real-time or supplied directly by video camera. Methods for analyzing video content and estimating resource requirements have been described in Chapter 2.

The content-based video traffic model takes into account the process of VBR stream generation. This provides the basis for more accurate traffic prediction and

	QoS of requesting stream	Renegotiation rejection probability
resources overestimated	increase or no change	higher
resources underestimated	decrease (loss)	lower

Table 4.1: Influence of prediction error on QoS and renegotiation rejection probability.

efficient video segmentation ultimately resulting in an increase of network utilization. The content-based video segmentation operates on a time scale of several seconds (corresponding to the scene length scale). The operation on this time-scale (VBR burst scale) is preferable to the frame-by-frame scheme. The frame-by-frame renegotiation could create a bottleneck due to the limited processing power at the switch or network interface.

Table 4.1 shows the influence of prediction error on stream QoS and rejection probability. QoS of the requesting stream (for example cell loss) might be increased or not effected if the traffic prediction model overestimates the amount of required resources. Nevertheless, since more resources are reserved than actually needed, the renegotiation rejection probability of other streams increases and overall network utilization decreases. On the other hand, decrease in QoS might be a result of the traffic prediction model underestimating the amount of required resources. In this case, the source traffic must be scaled. This can be accomplished in two different ways. Either the dynamic shaping can be applied to the already compressed stream in real-time, or an encoder will be notified to decrease the bit rate (e.g., to increase quantization step size). In addition, the probability of successful renegotiation of other streams will increase (e.g., there will be more resources available).

The content-based network resource allocation algorithm can be used in both real-time and non real-time systems. The main difference between both systems is that the non real-time system has all the information about scene content, including

the renegotiation points, scene length and resource requirements available before the stream is sent to the network. In addition, the off-line content-based segmentation is more accurate, since it does not experience delay associated with the determination of the video content. In off-line segmentation, resources can be requested ahead of time. The real-time system is more complex, since it assumes no previous information about the stream. The real-time system can be used for live video programs. On the other hand, off-line segmentation can be used in video-on-demand applications only.

4.3.1 Media Traffic Agent

Figure 4-4 depicts the content-based media traffic agent that manages the video stream transmission. The video segmentation and traffic prediction is based on the content-based video traffic model. The MTA monitors the incoming compressed video stream and extracts video content that is used for prediction of stream bandwidth requirements. MTA consists of five components: content analyzer/classifier, traffic analyzer, traffic prediction module, resource reservation module, and dynamic resource shaping module.

Module functions are as follows: the video stream is first analyzed by content analyzer/classifier, CA/C. The video stream is segmented into individual activity periods, content features are estimated and traffic class, $\mathcal{T}_i$, of the activity period is determined. In practice, there may be delay associated with the extraction of content features. The delay depends on the particular compression mechanism and the method that is used to determine content features. For example, the delay of several frames may be needed in MPEG-2 to determine spatial complexity directly from the compressed stream without fully decoding video stream. For that reason, the content-based MTA system is best suited for live video, which differs from

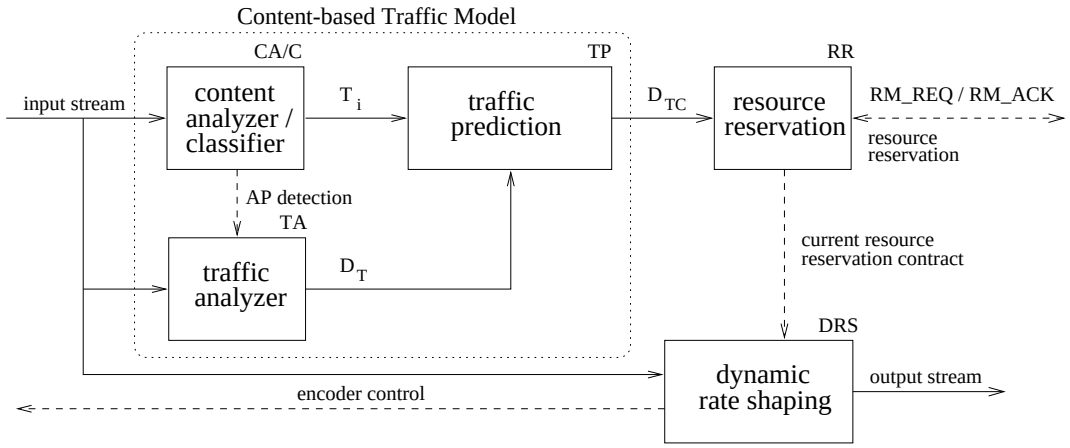

Figure 4-4: Content-based Media Traffic Agent (MTA).

interactive video in that it relaxes the end-to-end interactive delay requirement.

The traffic analyzer, TA, is triggered by CA/C at the beginning of each activity period. It continuously measures the traffic descriptor or each activity period. At the end of each activity period, both the activity period's traffic class and the traffic descriptor are used to update the characteristic traffic descriptors, $\mathcal{D}_{T,T_i}$, at the traffic prediction module.

The traffic prediction module, TP, predicts resource requirements for the current activity period. It maps the traffic class, determined by CA/C, into $\mathcal{D}_{T,T_i}$ descriptor. Values of $\mathcal{D}_{T,T_i}$ descriptors may be adaptively updated based on $\mathcal{D}_T$ descriptors of previous activity periods that were classified into the same traffic class.

The resource reservation controller, RR, decides whether to initiate a renegotiation of network resources. If necessary, a reservation request is sent to the RAB to either increase or decrease the currently reserved stream's bandwidth. If new resources are successfully obtained, the stream is sent to the network without any changes. However, if resources cannot be allocated, previously obtained resources have to be used. In that case, the source should be (i) regulated (e.g., dynamic rate

shaping), (ii) delayed or (iii) the stream may enter the network, but marked as a lower priority.

There is a tradeoff between the use of delayed transmission, regulation of the video stream and the use of the packet prioritization. Regulation of the source causes controlled, but immediate video degradation. However, it does not introduce significant delay. On the other hand, although delayed transmission does not cause any immediate quality degradation, it requires additional buffering and may not be suitable for interactive applications. The stream consisting of low priority packets may maintain the quality under the low network load. However, packets are randomly discarded at network nodes in case of congestion. In that case, uncontrolled quality degradation would take place.

If required, the stream is regulated at the dynamic resource shaping module, DRA, or directly at the encoder (in real-time by modification of its quantization parameters). A policing of D-BIND constrained sources can also be accomplished by a set of leaky buckets [54].

Content-based scheme is best suited for prediction of resource requirements for live video. For stored video, optimal segmentation can be computed directly from known video trace statistics, as described in [53, 54, 44]. In our simulations, the CB model compared favorably, in terms of prediction accuracy and renegotiation frequency, to other bandwidth prediction models. The performance was significantly better than other techniques for live video simulations.

4.4 Network Simulations

We compare the performance of content-based DRA and three other DRA schemes (i.e., frame-based, RVBR and RCBR) based on trace-driven simulations. A trace-driven simulator was developed for that purpose. Results were obtained using a

single 54000-frame-long trace (30 minutes) of an MPEG-2 encoded movie, Forrest Gump [43]. This trace includes over 300 activity periods of different content.

4.4.1 Trace-Driven Simulator

Figure 4-2 depicts a model of entry node multiplexer, with a service rate of c = 45 Mbps and FCFS scheduling policy. Given the average rate of our video stream (i.e., 0.5 Mbps), maximum of 90 sources would be possible to multiplex to assure a stable system. At the beginning of the simulation, each source was assigned a random starting point within the trace. When the source reached the end of the trace, it wrapped around and continued from the beginning of the trace. Each source dynamically requests resources at renegotiation points that were determined by the video segmentation algorithm of each given DRA scheme. For simplicity, it was assumed that once the request for more resources was rejected by RAB, the corresponding source was blocked and the new request was generated at the next renegotiation point only. As we will show later, it may be possible that resources, predicted at the beginning of the activity period, are underestimated. In that case, a new resource allocation request is generated. To limit the renegotiation frequency, in our simulations, we request a 10% increase of the currently needed (i.e., measured) resources.

In each simulation, the maximum number of streams that can be multiplexed was obtained. Experiments were conducted using renegotiation blocking probability of 10^{-2}. We assume shared network buffer size to be equivalent to 300 ms delay. For each result, simulations were run 100 times, starting at different random points within the trace. Performance evaluation was based on link utilization, defined as the ratio of the number of streams that are admitted under a given DRA policy to the maximum number of streams that are admitted under the average rate admis-

sion policy. The trace driven simulator was based on the following algorithm:

1. Fix buffer size and initial number of multiplexed sources
2. Run simulation 100 times and calculate rejection probability
3. If rejection probability is less than the maximum specified,
   increase number of sources and run simulation again from 2
4. Else, save results (number of sources)
5. If needed,
   select another buffer size and start from 1
6. Else, end simulation

4.4.2 Results

4.4.2.1 Performance of off-line video segmentation algorithms

In this section, we compare the efficiency of the following off-line video segmentation algorithms: (i) "frame-based", (ii) "APD manual" (manual content-based video segmentation, (iii) "APD auto" (automatic content-based video segmentation, and (iv) "RVBR off-line" scheme [54].

Since network efficiency is heavily influenced by the renegotiation frequency of DRA, parameters of "APD auto" and "RVBR off-line" segmentation schemes were tuned such that each algorithm produced a renegotiation frequency equal to that of "APD manual" segmentation, described in Section 3.3.3. The average interval between renegotiations for "APD manual" scheme was 3.3 s. Renegotiation frequency of "RVBR off-line" segmentation scheme is controlled by a single parameter ψ ($0 \leq \psi \leq 1$). The parameter, ψ, was adjusted to $\psi = 0.65$ such that the average interval between renegotiations was about 2.58 s, close to that of the "APD manual" segmentation scheme.

In this experiment, the actual activity period traffic descriptors ($\mathcal{D}_T$) of each activity period (e.g., D-BIND for all DRA schemes except "frame-based") were pre-computed off-line. In our simulations, $\mathcal{D}_T$ descriptors were used to generate resource requests at the beginning of each activity period.

Simulation results are shown in Figure 4-5. As expected, utilization of the "frame-based" scheme was superior; it established an upper bound on network utilization for a given video. High link utilization, the result of high statistical multiplexing gain, is due to a high frequency of renegotiations. As the network buffer (and corresponding delay) increases, utilization of other schemes is asymptotically approaching utilization of the frame-based scheme. While "APD auto" scheme has slightly better performance ($\approx 5\%$) for small network buffers, compared to "RVBR off-line" scheme, for large buffers their performance is almost undistinguishable. At large delays, "APD manual" scheme shows a small, 5% improvement of the utilization, compared to both "APD auto" and "RVBR off-line" segmentation schemes. Note that the effect of network buffering is present in all segmentation schemes. Network buffering is able to smooth the bit rate of the source, resulting in a further increase in network utilization.

Although we can conclude that in our simulations all segmentation schemes performed similarly, both "APD manual" and "APD auto" video segmentations have shown up to a 5% improvement, compared to "RVBR off-line" scheme. However, note the difference between "APD auto" and "RVBR off-line" schemes. The advantage of "APD auto" algorithm is that it can be used for real-time segmentation of video streams, while "RVBR off-line" scheme, which is not causal, cannot.

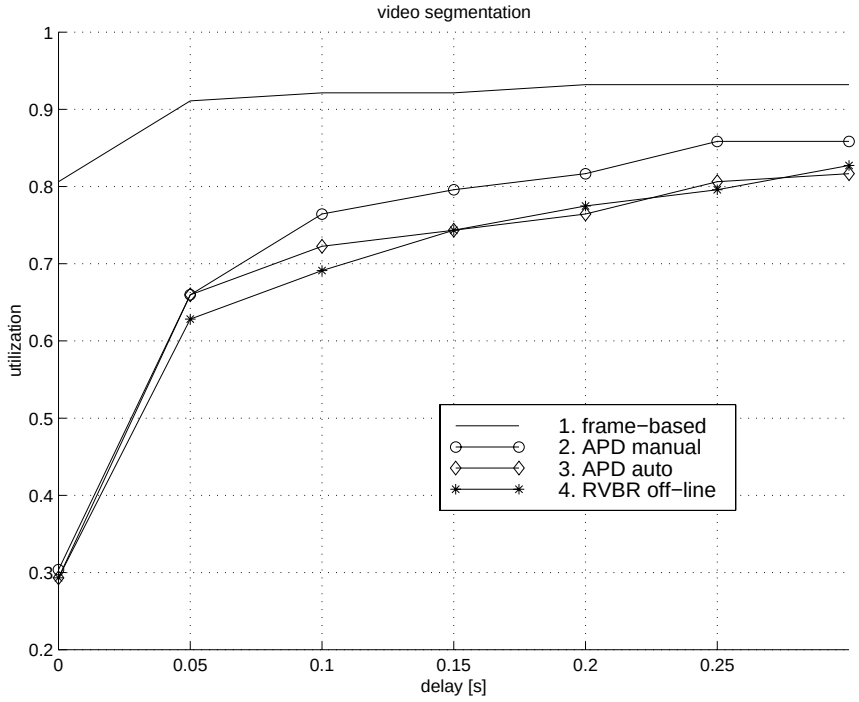

Figure 4-5: Impact of video segmentation accuracy (off-line segmentation).

4.4.2.2 Performance of DRA Schemes

In this section, we compare the performance of the following dynamic resource allocation schemes: (i) content-based resource clustered (CBRC), (ii) real-time renegotiated variable bit rate (RVBR), and (iii) real-time renegotiated constant bit rate (RCBR) [53].

In CBRC, bandwidth requirements of activity periods (i.e., $\mathcal{D}_T$) cannot be determined precisely for live video; characteristic traffic descriptors, $\mathcal{D}_{T,T_i}$, are used instead. In our simulations, activity periods were classified based on content features extracted from the video stream in the first group of frames (GoP) after the change of the activity period. The decision tree in the content-based analyzer/classifier (CA/C) and all $\mathcal{D}_{T,T_i}$ characteristic traffic descriptors in the traffic prediction module (TP) have been initialized before the start of simulations (refer to Section 3.4).

	MAX	PER	AVG
CBRC, Ideal CA/C	0.01%	1.1%	18%
CBRC	5.3%	8.1%	28%

Table 4.2: Influence of $\mathcal{D}_{T,T_i}$ evaluation metrics on renegotiation frequency of CBRC scheme (relative to "APD auto") for rejection probability=0.

Activity periods of MPEG-2 stream that were used for the training of the content-based classifier were not used for simulations. Note that in practical implementations, traffic classes can be updated (i.e., continuously learned) as new video activity periods arrive.

In our simulations, we have used three different evaluation metrics for computation of characteristic traffic descriptors, $\mathcal{D}_{T,T_i}$. The descriptors were computed as average (AVG), 90th percentile (PER), or maximum (MAX) of all traffic descriptors, $\mathcal{D}_T$, of activity periods that were classified into the same activity class. The computation of $\mathcal{D}_{T,T_i}$ using the maximum metric corresponds to the deterministic case. In other words, bandwidth requirements of any activity period, classified into a particular activity class, never exceed predicted resources. Of course, this ideal case can be achieved only under the assumption of an ideal content-based classifier that produces no error in content classification. The deterministic evaluation (e.g., using maximum metrics) creates conditions, similar to peak rate allocation, under which network resources are underutilized. Unless we indicate otherwise, in the following, we assume MAX evaluation metrics for CBRC scheme.

On the other hand, resource requirements of activity periods for which $\mathcal{D}_{T,T_i}$ was computed using AVG or PER metrics may, in some cases, exceed predicted resources. If the current resource requirements exceed reserved resources, it may be necessary to generate an additional resource allocation request. Alternatively, the stream can be delayed, shaped (e.g., DRS), or it may enter the network, but marked as a lower priority. The DRS can be used to limit the source traffic such

that it conforms to the previously obtained $\mathcal{D}_{T,T_i}$ descriptor. DRA causes controlled, but immediate video degradation. On the other hand, a stream consisting of low priority packets may be discarded at network nodes only in the case of congestion. Correspondingly, if congestion does not occur, underestimation of $\mathcal{D}_{T,T_i}$ descriptors may result in an increase in network utilization.

The underestimation of resources may also occur in the case of non-ideal CA/C; an activity period may be incorrectly classified by the decision tree classifier. For example, the increase of the renegotiation frequency of our MPEG-2 stream for different evaluation metrics and the ideal or non-ideal CA/C is shown in Table 4.2. It is assumed that all requests are accepted (i.e., zero rejection probability). In our experiments, the accuracy of the decision tree classifier is 86.14%. However, this non-ideal CA/C causes only a 5.3% increase in the renegotiation frequency for MAX metrics.

Figure 4-6 compares the results of six CBRC simulations. The simulations are based on the MAX, AVG, and PER evaluation metrics, described above. In addition, the results based on ideal and non-ideal CA/C are shown. Corresponding to these simulations, Table 4.3 and Table 4.4 show the average time between renegotiations and an increase in renegotiation frequency (relative to original "APD auto"). The results of our simulations show a tradeoff between different evaluation metrics and the renegotiation frequency. Although the AVG case achieves a 27% increase in utilization, its renegotiation frequency is 40% higher than the original. On the other hand, the increase of the renegotiation frequency due to the non-ideal CA/C is only 10%, for the MAX metrics. We have found no substantial difference in utilization between the non-ideal and ideal CA/C. However, the non-ideal CA/C causes an increase of renegotiation frequency for all evaluation metrics.

Figure 4-7 depicts the main result of our simulations: "APD auto", CBRC,

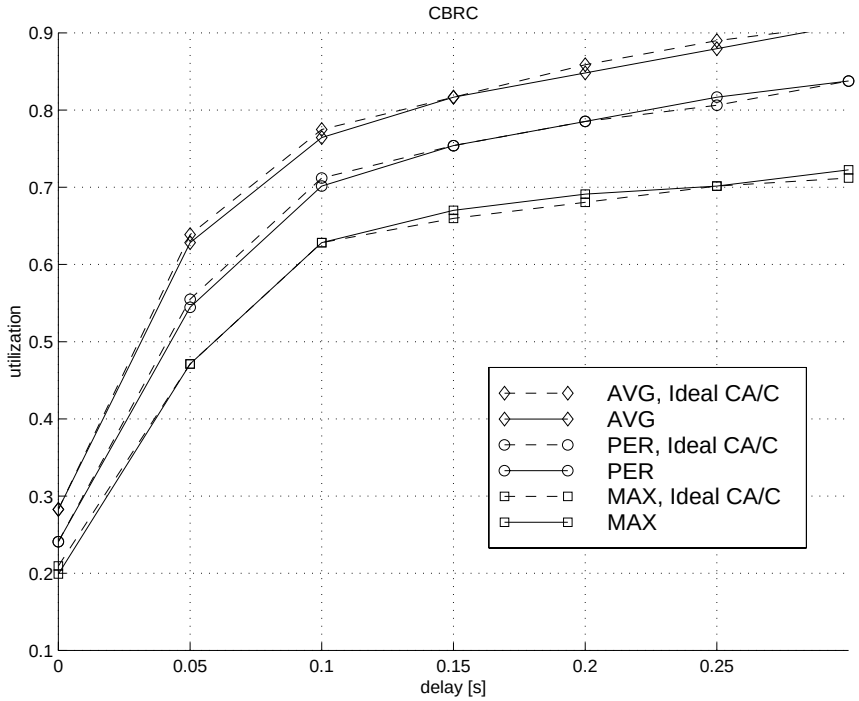

Figure 4-6: Effectiveness of CBRC schemes.

Evaluation metrics	Average time between renegotiations	Increase of renegotiation frequency
MAX	3.17 s/req	4%
PER	3.05 s/req	8%
AVG	2.54 s/req	29%

Table 4.3: Influence of $\mathcal{D}_{T,T_i}$ evaluation metrics on renegotiation frequency of CBRC scheme (relative to "APD auto"), ideal CA/C.

Evaluation metrics	Average time between renegotiations	Increase of renegotiation frequency
MAX	2.98 s/req	10 %
PER	2.85 s/req	15 %
AVG	2.34 s/req	40 %

Table 4.4: Influence of $\mathcal{D}_{T,T_i}$ evaluation metrics on renegotiation frequency of CBRC scheme (relative to "APD auto").

RVBR, and RCBR dynamic resource allocation schemes for three different renegotiation frequencies. The first curve, "APD auto", serves as a benchmark. It corresponds to the automatic activity period detection combined with precise off-line $\mathcal{D}_T$ descriptor evaluation (also shown in Figure 4-5 as "APD auto".) The second curve corresponds to the CBRC scheme, described previously (shown in Figure 4-6 as MAX). For comparison, three other simulations of RVBR scheme are shown. In RVBR, parameters α and β control the renegotiation frequency [54]. Three different parameter sets were used for each simulation: ($\alpha = 1.1$, $\beta = 0.9$), ($\alpha = 1.2$, $\beta = 0.8$), and ($\alpha = 1.3$, $\beta = 0.7$) resulting in 1.15, 2.23, and 4.23 s/request mean interval between renegotiations respectively. Real-time RVBR video segmentation is described in more detail in Section 4.2.2.2. For comparison, performance of three RCBR schemes with 1.15, 1.44, and 2.0 s/req mean intervals between renegotiations are shown.

Best overall performance was achieved with "APD auto" scheme, which has shown a sharp increase in utilization from 31% (no buffering) to 81% at the buffer of 20 Kbytes/stream. It is apparent that the performance of both RVBR and RCBR schemes depend extensively on renegotiation frequency.

Network simulations have shown that the link utilization achieved by all three on-line RVBR schemes was substantially less than utilization achieved using the "APD auto" (about 55% - 70% difference). Compared to the CBRC scheme, only one RVBR scheme with substantially a higher renegotiation frequency (corresponding to 1.15 s/request) achieved higher utilization at low buffer sizes. Two other RVBR schemes (corresponding to 2.23 and 4.23 s/request) achieved network utilization that was 15% - 20% lower. Average renegotiation frequency of CBRC scheme was 2.98 s.

The RCBR scheme has shown very low utilization at small buffer sizes per

stream, but its utilization increased sharply at large buffer sizes. This effect and the low performance of the RCBR scheme is explained by its separate buffering (refer to Section 4.2.2.3).

The superior performance of the CBRC, based on the content-based model, can be attributed the fact that it is able to track changes in video content (and therefore changes in the bit rate) better than other considered schemes. This is accomplished mainly by detecting natural discontinuities in video content. Second, traffic prediction based on the video content may improve prediction accuracy compared to schemes that use only the bit rate and network buffer occupancy in their heuristics segmentation and resource prediction algorithms. However, the most distinguishing feature of CBRC is its lower renegotiation frequency, compared to RVBR and RCBR schemes.

4.5 Remarks

Traffic prediction based on video content is a new approach that is suitable for live as well as stored video. It is directly applicable to dynamic resource allocation systems. The content-based model uses the video content of the video as an important indicator of VBR bandwidth requirements. In addition, the content-based model can also be used for off-line synthetic traffic generation.

The main advantages of a content-based framework are (i) lower renegotiation frequency and (ii) reduction of network resources. In our trace-driven simulations of a dynamic resource allocation system, a significantly lower renegotiation frequency ($\approx 60\%$ decrease) was needed to achieve similar network utilization for the CBRC system that was based on the CB model, compared to other systems based on existing DRA approaches (e.g., RVBR and RCBR). The CBRC scheme achieved a significant reduction ($\approx 55\%$ to 70%) in network resources under conditions of equal

renegotiation frequency.

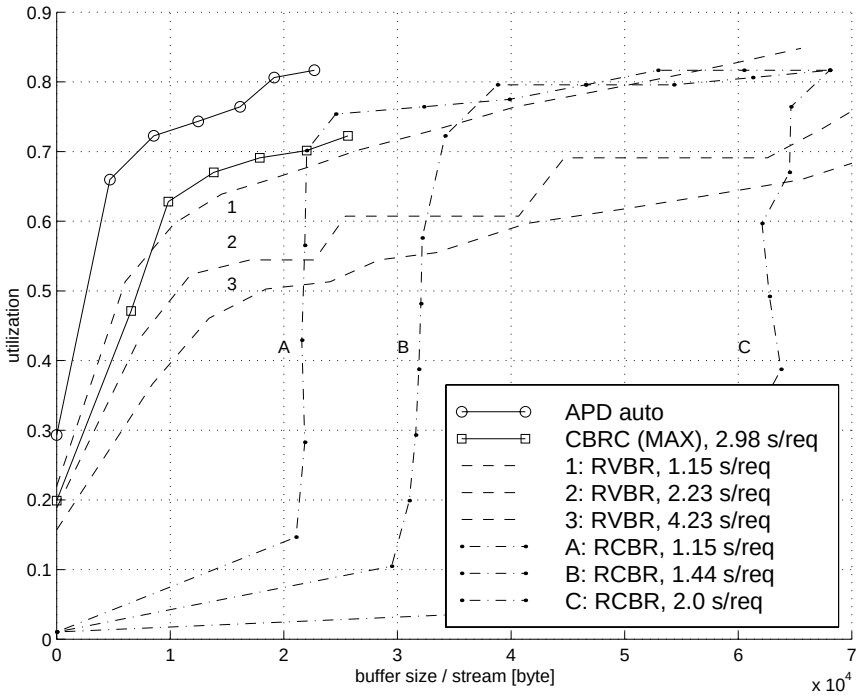

Figure 4-7: Effectiveness of CBRC, RVBR, and RCBR schemes.

Chapter 5

Media Scaling and Adaptation

5.1 Introduction

Currently, system support for multimedia has already become integrated into most workstations, PC's, and other systems. However, quality of service (QoS) requirements of many multimedia applications limit their potential functionality, especially in mobile and wireless environments.

Major changes to the wireless infrastructure have to be accomplished before multimedia services could be deployed over wireless networks. Compared to wireline networks, quality of service (QoS) delivered by wireless networks is influenced by many additional factors. Time-varying bandwidth characteristics are a result of fading, shadowing, path loss, cochannel interference, etc. Besides these limitations imposed by the physical layer, available bandwidth also depends on users' service and traffic requirements, mobility, location, handoff frequency and system-specific implementations including media access protocol and error correction algorithms [18]. Given finite spectral bandwidth assigned to mobile communications, wireless links can be characterized, unlike traditional wireline links, as time-varying bandwidth-limited channels. The QoS and bandwidth delivered to end-user applications is highly time-varying as well. Naturally, network efficiency, as an economic

factor, plays an important role in the wireless multimedia communications systems.

It is unlikely that given the physical conditions in wireless networks, hard QoS can be guaranteed. Over the past several years, it has been studied that video transport at wireless networks can be efficiently addressed using multimedia adaptation. In essence, multimedia applications should dynamically adapt their content to changing network conditions. Similarly, the networks should use appropriate media scaling policies and resource allocation techniques to compensate for network conditions in order to satisfy specific needs of applications. Although the adaptation framework is applicable to multimedia in general, in the following, we will focus on natural video only.

Media adaptation (MA), content-based scalability (CS), dynamic resource shaping (DRS) and dynamic resource allocation (DRA) are novel techniques well suited and directly applicable to the transport of video in conditions of time-varying bandwidth. It is generally accepted that video quality can be improved by exploiting video content scalability through rate control coupled with media scaling techniques [84]. In addition, as we have shown in Chapter 4, a significant increase of network utilization can be achieved using content-based dynamic resource allocation.

Due to the time-scale mismatch between application adaptability and variability of network conditions, it is instrumental that dynamic adaptation is accomplished in not only the application layer, but also other layers. Recently, several works presented adaptation policies and resource allocation algorithms in transport, network and data link layers [16]. For example, the Utility-fair Resource Allocation scheme introduced in [76] proposes adaptation in the wireless data link layer. The system uses a centralized adaptation controller and distributed adaptation handlers. The adaptation scheme is based on utility functions (UF). Utility functions may be used for resource management, bandwidth allocation and scheduling of scalable streams

sharing the network resources [20]. A discussion of corresponding networking protocol can be found in [81].

Besides the adaptation at end nodes, media adaptation can be accomplished in network nodes as well. For example, the utility function may be transmitted along with the compressed video stream to network nodes in the connection path where it can be used for resource allocation. At network nodes, media scaling agents adapt the streams accordingly, in the case of bandwidth variations.

The utility function is well suited for content-based adaptation and a media scaling framework. Although previous studies related to QoS support in wireless or bandwidth-limited networks indicated the importance of utility function as a soft media scaling indicator [71, 85, 81], little attention was given to the generation of utility functions. For stored-video, the utility function can be estimated offline. This option is generally not available for live video. More importantly, direct estimation of utility functions may require extensive computation. New methods that simplify the estimation of utility functions are necessary.

In this chapter, we present a novel framework for utility function estimation that may be used for both stored and live video encoded by standard encoding methods such as MPEG-2 and MPEG-4. In Section 5.2, we discuss media scaling issues including video quality metrics and utility functions. In Section 6.1, we introduce a novel framework for utility function estimation. In particular, we use video content to classify video objects into different utility classes, each of which is associated with distinctive utility functions. In the last section, we discuss our experimental results of content-based approaches to utility function generation.

5.2 Media Adaptation

Media adaptation can be achieved at different layers, however, application and session layers are more appropriate for considering user-perceived quality. Adaptation can be accomplished at various points in the network: at the video source, receiver and network nodes. In general, the media scaling techniques can be categorized as: (i) spatial resolution scaling (e.g., change of picture size), (ii) temporal domain scaling (e.g., frame dropping), (iii) quality scaling (e.g., change of quantization levels, chrominance dropping, DCT coefficients dropping, etc.), and (iv) content-based scaling (e.g., video object prioritization, adaptation and dropping).

Techniques applicable for media adaptation can widely vary, depending on the location at which they are used. For example, video streams of different resolutions can be selected at the source node to match the long-term bandwidth contracts. On the other hand, for short-term bandwidth limitations or variations, it will be more appropriate to apply dynamic rate shaping. In the network, other simplified media scaling methods (i.e., frame dropping, DCT coefficient dropping, etc.) can be used to react to temporary bandwidth variations.

MPEG-2 standard currently supports three-layer (base, medium, high) scalable encoding. First, video is encoded as a base layer. Medium and high layers can be used to further improve video quality of the base layer. In MPEG-2, the following scalability options are supported: spatial (size), SNR (quality), and temporal (frame-rate).

Three-layer scalability of MPEG-2 can be used in both stored and networked video applications. For example, based on receiver resolution or network conditions, streams corresponding to different scalable layers can be multiplexed and transmitted over the network. By receiving additional medium or high layer streams, a receiver with higher bandwidth and/or resolution will be able to achieve higher

video quality. However, a small number of scalable layers (three in MPEG-2) limits its usage in networks that cannot sufficiently guarantee QoS. For these networks, finer-grain media scaling is more appropriate.

Finer granularity of scaling can be achieved, for example, during the encoding process, by varying quantization parameter, frame rate, image size, etc. For stored compressed video, this operation, called transcoding, may need additional computing resources because the encoded stream has to be first transformed back to the original spatial domain, and then re-encoded. However, delay associated with the transcoding operation can be compensated by retrieving the video ahead of playback.

Most scaling actions generate coarse-grained rate changes that can be estimated by the amount data dropped as "frames/layers/objects". The resulting distortion function will have a discrete drop for the scaled-down rate. In contrast, fine-granularity rate changes can be achieved by dynamic rate shaping (DRS) [83, 86] method. Although its performance, in terms of perceptual video quality, may be lower than that using transcoding, it does not require full decoding of the video. The fine-grain scalability of DRS is achieved by optimized DCT coefficient dropping. Compared to transcoding, this operation is relatively simple and may be accomplished in real-time in both sources and network nodes.

5.2.1 Content-based Scalability

The above-described media scaling methods operate at the frame-level. However, sometimes it is desirable to support even finer granularity of scaling. Content-based scalability is one of such techniques. It allows for scaling of specific regions of the frame independently.

In general, the content-based scalability can be accomplished at two levels: across

multiple media objects and within media objects. Given resource constraints, the resulting perceptual quality of the video at the receiver can be maximized by the assignment of different resolutions to various regions of the frame. Based on relative importance of video objects in the scene, their priority and content characteristics, an appropriate amount of resources (e.g., bandwidth) can be assigned to individual objects using utility-based resource allocation algorithms. For example, in some applications, it may be beneficial to preserve the resolution of the most important video objects (i.e., foreground) while reducing the resolution of the background object. Also, content characteristics of the video objects may influence the allocation of resources. For example, it may be beneficial to scale down the spatial resolution of fast moving objects because their details are not as important as details of stationary objects. However, the temporal resolution (frame rate) of a fast moving object is important and should be preserved. In contrast, slow motion objects (e.g., background) may be temporarily re-scaled such that they keep their spatial resolution while lowering their temporal resolution.

It is important to note that the combination of scaling techniques applied to each individual media object can vary as video content changes.

5.2.2 MPEG-4

MPEG-4 is a new video compression standard based on the object concept described above [92]. The most distinctive feature of MPEG-4 is an efficient representation of audio-visual objects of arbitrary shape. It supports most of the functionality already provided by MPEG-1 and MPEG-2 in its core video compression methods [90, 91]. One of the most notable new features supported by MPEG-4 is separate encoding of video objects appearing on the scene. This new feature allows advanced media scaling, namely content-based scalability. In addition, MPEG-4 was designed

according to requirements of both storage and communication applications. New error-control functions of MPEG-4 may improve the performance in low bit-rate and high bit-error rate wireless channels [17].

The MPEG-4 video stream can contain a number of MPEG-4 elementary streams corresponding to different media objects with distinctly different adaptation requirements. This way, content-based scalability can be efficiently combined with other mechanisms that influence visual quality, spatial and temporal resolution. In fact, each individual video object in MPEG-4 can be independently scaled to the point that it is replaced by a static icon, or it is completely removed. At the result, the total bit budget assigned to the video scene (e.g., scene containing more than one video objects) may be effectively distributed among individual video objects such that the optimal subjective quality is achieved.

For simplification, in the following, we use the term "video object" synonymously to refer to either video frame (for frame-based encoding schemes such as MPEG-1, MPEG-2, H.263, etc.) or video object plane (VOP) in MPEG-4 terminology.

5.2.3 Video Quality Metrics

Video compression causes quality degradation of video content. User's satisfaction with the quality can be measured by various "quality indicators". The simplest quality indicator is a binary (e.g., "yes/no") indicator that expresses only satisfaction/dissatisfaction with currently offered quality of service [71]. The measure of quality can be objective or subjective. Based on application requirements, quality can be defined on a linear or discrete scale. For example, linear objective quality measure can be defined as SNR (signal to noise ratio), PSNR (peak signal to noise ratio), and WSNR (weighted SNR). Although objective quality does not correspond well to visual quality perceived by humans, its estimation is relatively simple. It

was used in many image compression applications including rate control.

In contrast, subjective quality models can achieve results that are more accurate in terms of human perception [19]. However, high computational demand limits their use in real-time applications. One example is "Picture Quality Assessment System", originally developed by Hamada [73]. It was reported that this system achieved results that were highly correlated (0.8 - 0.9 correlation) with ITU standardized subjective tests with 20 viewers. The experiments were conducted on a hardware-assisted stand-alone device that was able to compute the picture quality of MPEG-2 compressed video streams in real-time. The quality estimation process was based on the human visual system and modeled at three basic layers: object, texture and local noise. The distortion residuals computed as a difference between original and encoded pictures were first weighted based on local image characteristics at each layer, and then aggregated at higher layers. At the resultant object layer, the weighted noise was summed over the whole image and converted into a distortion scale and a 5-level satisfaction index called mean opinion score (MOS), recommended by ITU-R Rec. 500-7 [74]. MOS can be regarded as a simple and intuitive measure of subjective video quality ranging from 1 (unsatisfactory) to 5 (excellent).

5.2.4 Utility Function

In network engineering, utility functions represent a powerful framework for characterizing the ability of applications, or media, to adapt to varying network conditions. Specifically, in the context of bandwidth allocation, utility functions indicate achievable quality as a function of available bandwidth. As described above, utility functions can be based on subjective or objective quality metrics. Utility functions have been successfully applied in several network resource allocation algorithms [71, 85, 81]. In addition to adaptation and network resource allocation, subjective

utility functions can be efficiently used for bit rate control in the encoder and for dynamic rate shaping (DRS) [83] at network nodes.

In general, utility functions have a functional form depending on many variables (e.g., power, memory, etc.). In video networking applications, the utility function is typically defined as function of a single variable: bandwidth. In this case, the utility function represents the relationship between the user's satisfaction index and the network bandwidth. The MOS-based utility function allows simple and efficient expression of media scalability.

The subjective quality of video scenes can be estimated from the subjective quality of each individual video object and its priority using "utility-fair resource allocation" algorithms [76]. In this case, priority refers to the relative importance of the object in terms of visual quality as perceived by the user. In practice, video object priority may be determined based on heuristics that take into account its content features (e.g., object position, speed, complexity, etc.). Alternatively, for off-line video, the video object priority can be explicitly indicated manually during the video editing process.

Estimation of utility functions requires repetitive computation of quality under different resource conditions. Usually, it requires measurement of video quality while varying different coding parameters (e.g., quantization steps, frame rate, etc.). In this case, utility functions may be specified in parameterized form, using a limited number of sampling points. Values between sampling points of parameterized utility function can be, if necessary, computed by interpolation. The process of repetitive estimation is the main cause of the large amount of calculations required for a utility function evaluation.

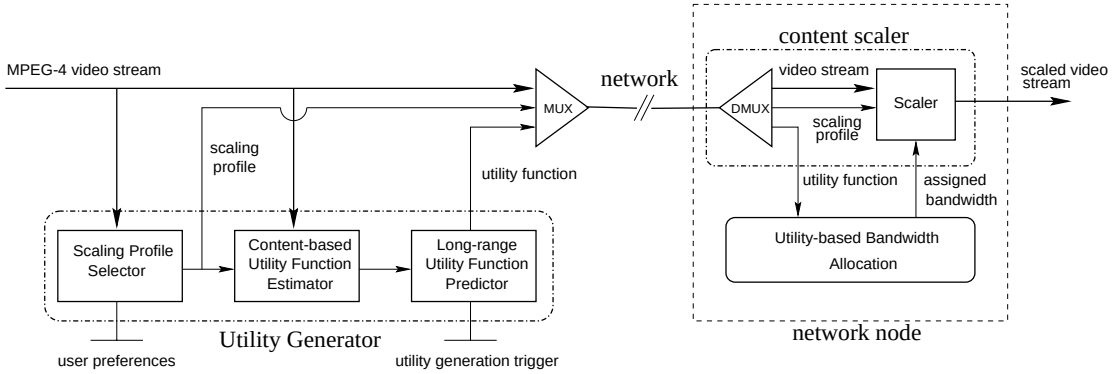

Figure 5-1: Conceptual model of the media scaling network.

5.3 Conceptual Model

The conceptual model of the media scaling network is illustrated in Figure 5-1. It depicts three modules that constitute the utility generator including a scaling profile selector, content-based utility function estimator and a long-range utility function predictor.

A scaling profile selector addresses the content-based scalability function, including an aggregation of multiple media objects and the selection of potentially multiple scaling techniques for a single media object. The scaling profile selector uses information about user scaling preferences and content features to select the appropriate scaling profile for the video stream. The role of the scaling profile is to ensure that a content scaler, which can be located inside the network (refer to Figure 5-1), will select the same combination of scaling methods used by the utility function generator for the reduction of the bit rate.

Utility functions and scaling profiles are dynamically created by the utility generator and dispatched to the content scaler as illustrated in Figure 5-1. The content scaler forwards utility functions to the bandwidth allocation module to make resource reservations. Since the generation of utility curves and scaling profiles can occur frequently, an efficient signaling scheme is required to ensure a timely delivery

of scaling information inside network. One of the options is to use the MPEG-4 or MPEG-7 object descriptor structures [92, 93] to update utility functions and scaling profiles at the content scaler, located at the network nodes.

A scaling profile captures the scalability of a video at two levels: (i) the scaling pattern of single video objects (i.e. combination of scaling techniques applied to one object); and (ii) the aggregation of multiple prioritized video objects in the same video stream. For example, in MPEG-4, a number of elementary object streams corresponding to different video objects can be multiplexed into the same network session [105]. In that case, utility functions constructed for single video objects need to be aggregated together into the form suitable for the utility-based bandwidth allocation. In the case of aggregation, a video object priority may be used to define the scaling order for different video objects: a low priority video object will be dropped first, before a high priority object is scaled down. Objects that have the same priority will be scaled proportionally, for example, using the utility-fair algorithm [81].

The combination of scaling techniques applied to a video object is not static, but dynamically changing and based on user preferences and video content. For example, for a fast-motion scene, spatial resolution or quality scaling methods are more suitable than temporal-domain scaling techniques (e.g. dropping frames) because the details within a picture may not be important under fast-motion. However, slow-motion scenes favor the opposite approach. In the architecture illustrated in Figure 5-1, the scaling profile selector is co-located with a content analyzer, gaining access to video content features. In addition, the module provides a user interface to specify high-level rules used to generate scaling profiles (e.g., a mobile PDA user may prefer high resolution to rich color). User preferences may be specified, for example, as a sequence: first, dropping chrominance, second, dropping background

objects and, third, reducing frame rate of foreground objects.

The utility function generation requires repetitive measurement of video quality distortion. The procedure can be computationally intensive if fine-granularity sampling is taken. To reduce the impact of online utility generation on the transport system, the generator is (i) architecturally separated from the packet forwarding path and (ii) the utility generator is placed inside the video server. In addition, as we will show later, by applying machine learning techniques, we can lower an amount of computationally intensive procedures of the utility function estimator so that the generation of the utility function is possible in real-time.

The utility functions generated for video may dynamically change over the fast time-scale (e.g., in the order of tens of milliseconds) as the content changes (e.g., due to scene changes in a video stream). An important observation from the network resource management point-of-view is that network adaptation operates over a longer time-scale (in the order of hundreds of milliseconds to tens of seconds). This is a product of the signaling system efficiency, resulting network load of dynamic resource management systems and the round trip delay between a source encoder and receiving decoder. Since the "content variation time-scale" may be several orders of magnitude smaller than the "network adaptation time-scale", the utility generator needs to reconcile the potential mismatch in time-scales, but without significantly sacrificing the accuracy of the generated utility function, or burdening the network with large volumes of signaling.

The time-scale mismatch discussed above is addressed by a long-range utility function predictor. This module uses an adaptive filtering algorithm [82] to keep the generated utility functions stable over a long network adaptation time-scale. The algorithm adjusts itself to track the long-term variation in utility functions to balance the tradeoff between increasing the utility generation interval and maintaining

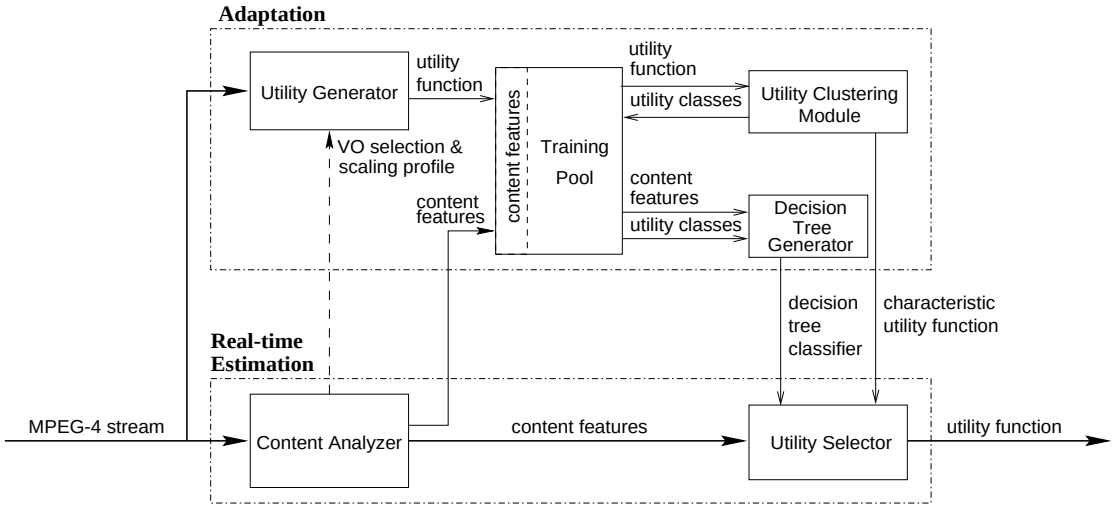

Figure 5-2: Content-based utility function estimator.

the accuracy of the generated utility function.

In the next section, we will discuss the detail design of the content-based utility function estimator module, illustrated in Figure 5-2

5.4 Content-based Utility Function Estimator

The dynamic generation of utility functions is time-consuming and requires a large amount of processing power. Given the current technology, generation of utility functions on a frame-by-frame basis is difficult to achieve in real-time.

The proposed solution is based on the use of video content. In previous chapters, it has been shown that content features relate directly to bandwidth requirement. These findings constitute a basis of our system. We show that, similar to bandwidth requirements, content features also relate to the shape of the utility function. In what follows, we describe a technique for content-based utility function estimation that accelerates the generation of utility functions.

The proposed acceleration technique does not explicitly compute utility functions for each video object. Rather, machine learning techniques are used. The system

uses video content, represented by a set of content features, to determine the utility class of the object. Because the video content can be dynamically extracted from compressed video streams, this technique is suitable for real-time applications.

Figure 5-2 illustrates the architecture of a content-based utility function estimator that can support a variety of compression schemes (e.g., MPEG-1, MPEG-2, MPEG-4, H.263, etc.). The system architecture comprises two main components: a real-time estimation module and an adaptation module. The adaptation module comprises the utility generator, training pool, utility clustering module, and decision-tree generator. The real-time estimation module comprises the content analyzer and the utility selector.

The architecture of the system allows smooth adaptations during continuous operation of the system. The adaptation module, which is computationally intensive, is decoupled from the real-time estimation module. Both modules operate asynchronously as follows: during the normal operation, based on content features extracted online by the content analyzer, the utility selector dynamically determines utility class and select corresponding characteristic utility function for each video object. The characteristic utility function is used as an estimator of real utility function that is not explicitly computed for each video object. An adaptation module is activated periodically to re-compute the decision tree parameters used to implement the utility selector. In this manner, by avoiding explicit per-object generation of utility functions, the system facilitates operation in real-time.

5.4.0.1 Content Analysis

For illustration, we present an operation using MPEG-4 video streams. Before entering the content analyzer, MPEG-4 video programs are demultiplexed and individual video object streams are extracted. The analyzer processes video object streams and

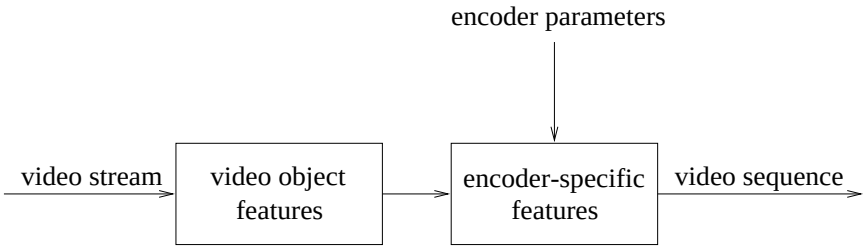

Figure 5-3: Video content features.

extracts video content information in real-time [59]. Content information comprises visual features and encoder-specific features (Figure 5-3). Visual features describe video object characteristics (e.g., video object size, speed, etc.) that do not change if an alternative encoding technique is applied. In contrast, encoder-related features are sensitive to specific encoding technique and encoder parameters (e.g., frame type, DCT values, number of bits for various encoder-specific stream components, etc.). Most content features can be extracted directly from compressed video streams (e.g., object size, frame type).

Extraction of the content features is somewhat dependent on the location of the content analyzer in the system (e.g., at the video server or network nodes). When content analyzer is not co-located with the video server, information that cannot be obtained from video streams directly can be carried in the extension fields of MPEG-4 or MPEG-7 object descriptors [92, 93]. In this case, the estimator may be placed at network nodes.

Accurate estimation of video object features is complex and requires substantial processing power. Video content can be accurately evaluated at the original uncompressed domain. Alternatively, content features can be estimated with approximation directly from compressed video streams minimizing the need for intensive computations. Since high estimation accuracy of features is not crucial for most networking applications, estimation of content features in the compressed domain

is sufficient.

An additional function of the content analyzer is to select some video objects for further evaluation in the adaptation module. In general, the video object indicating a substantial change in its visual content will be selected. For example, if two video objects that are extracted from different scenes have content features substantially different from each other, these objects will be selected for evaluation in the adaptation module.

5.4.0.2 Real-time Estimation

The real-time estimation module allows the estimation of utility functions on a frame-by-frame basis. Estimation is based on the content features extracted by the content analyzer. Assume that the system has been initialized and the adaptation module has estimated the decision tree parameters of the utility selector. The real-time estimation of the utility function is realized in the following manner.

1. Content features are extracted in real-time from the compressed video stream by the content analyzer.

2. The utility selector uses these features to determine the utility class of the current video object. The selection is based on the decision tree that is periodically updated in the adaptation module.

3. Once a utility class has been determined, the characteristic utility function for that class (that is determined initially during the training period or during the adaptation period) is selected for the video object. The characteristic utility function is estimated from utility functions within the same utility class and is used as a utility function estimator for the class.

5.4.0.3 System Adaptation

The adaptation module dynamically computes parameters of the decision tree classifier used in the utility selector. The operation of the adaptation module is computationally intensive. However, it is invoked only intermittently and is functionally independent of the real-time estimation path. The adaptation module involves a set of video objects in the training pool that have been placed there, as explained earlier, by content analyzer. Each video object in the training pool is described by (i) the utility function and (ii) content features and (iii) utility class. Utility function is computed explicitly at the utility generator. Content features are extracted at the content analyzer. Utility class is obtained from the utility clustering module. Smooth system adaptation is assured by continuous replacement and intermittent re-clustering of objects placed in the training pool, as explained later.

The utility generator computes utility function for each video object in the training pool. The computation is based on the scaling profile [97]. The scaling profile provides information about user-preferred and network-supported scaling methods and their particular sequence. For example, the scaling profile may indicate, first, to use DRS to scale the bit rate to 30% of its original rate and, second, to use the frame dropping to further decrease the bit rate. The scaling profile is also used by media scaling agents. The function of media scaling agents, located inside the network or network interfaces, is to adapt the video stream to network conditions. This way, it is assured that utility functions, generated by utility generator, correspond to scaling actions that are applied in real-time by media scaling agents. Note that the utility generator may need to use the original video stream for the computation of the utility function. If the original video stream is not available, the utility function may be determined relative to the current video quality.

A utility clustering module performs automated clustering of video objects in

the training pool. The clustering is based on unsupervised classification algorithms operating on selected features of the utility function [78]. Utility functions of video objects are used as feature vector. In its operation, the utility clustering module uses only parameters describing utility function. After the completion of the clustering operation, each video object in the training pool is marked according to its utility class.

The decision-tree generator starts its operation after utility function clustering is completed. The generator is also based on machine learning techniques. However, compared to clustering module, supervised classification is used [79]. The decision-tree generator determines the decision tree by using (i) utility classes derived by the utility clustering module and (ii) content features, extracted by the content analyzer. At this point, the decision-tree generator does not use parameters describing utility functions. Instead, utility class and content features are used. This way, once the decision tree is formed, the utility class of the particular video object can be determined from the video object's content features only. This operation is performed by the utility selector. Once the utility class is determined, its characteristic utility function is used as an estimator of real utility function.

5.5 Evaluation

In the following, we present experimental results focusing on the effectiveness of the content-based utility estimator. Two experimental classification schemes implementing utility clustering module, the decision tree generator and the utility selector demonstrate the viability of content-based approach. The first experiment is based on MPEG-4 video. The two structurally different utility estimators suitable for MPEG-4 video are compared. In the second experiment, accuracy of the MPEG-2 content-based utility estimator is evaluated.

In all experiments, a randomly selected half of video objects available was used for training; i.e., generation of a decision-tree. A second set of video objects was used to obtain the classification accuracy of the utility selector.

5.5.1 MPEG-4 Content-based Utility Function Estimator

For this experiment, we have prepared seven MPEG-4 video object streams in the following way. Original video sequence was concatenated from 38 high quality CIF video shots of 100 to 300-frames long each. First, each shot was segmented into two or more arbitrary-shaped video objects using a system for semi-automatic video segmentation [77]. Each video object corresponds to the real object (e.g., person, airplane) in the scene. Second, each segmented video object was encoded using MPEG-4 VM8 software [80]. Finally, the resulting streams corresponding to video objects were concatenated to form a single 10734-frame long video object stream. Seven video object streams were prepared, each corresponding to a different DCT quantization coefficient $q = 1, 5, 10, 15, 20, 25, 31$.

Figure 5-4 shows seven traces of this original video sequence. From top to bottom, sub-graphs corresponds to streams encoded with different quantization parameters $q = 1, 5, 10, 15, 20, 25, 31$ respectively. In this figure, bit-rate is shown on the left sub-graphs and corresponding subjective quality is shown on right sub-graphs of the Figure 5-4. For example, the top-left graph depicts rate variations in bits/pixel of the stream encoded with $q = 1$. A video stream that is encoded using a small quantization parameter results in high bit-rate and high subjective quality. This is illustrated at the top-right graph of Figure 5-4, by constant utility estimated at level 5 for the entire length of the 500-frame period. On the other hand, bit rate variations that are shown at the bottom-left graph of Figure 5-4 correspond to a stream encoded with quantization parameter $q = 31$. In this case, although the

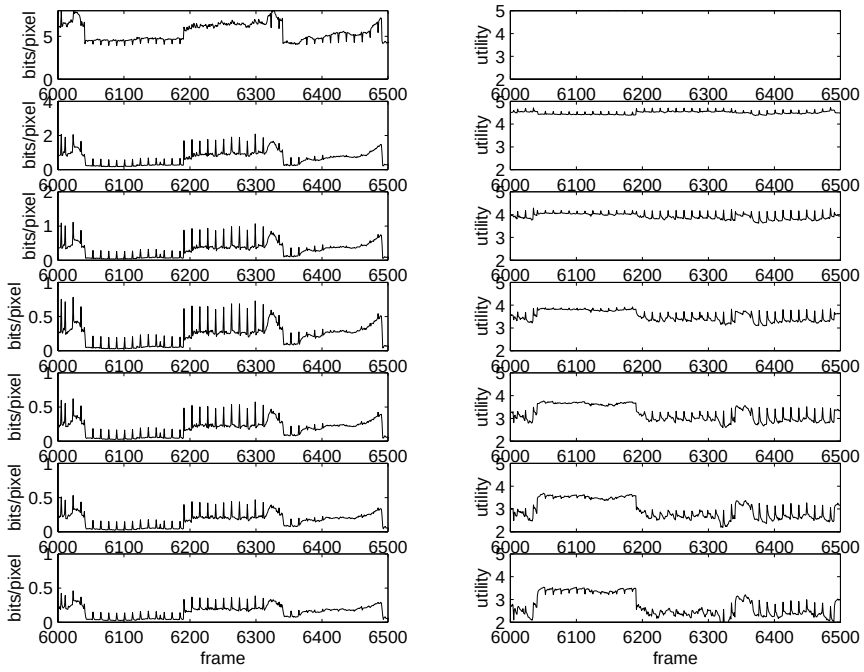

Figure 5-4: MPEG-4 stream bit-rate and corresponding utility estimation.

video sequence was encoded as VBR, constant quantization parameter did not result in constant subjective quality. This is illustrated at the bottom-right graph. Figure 5-4 depicts an important observation: encoding with a fixed quantization coefficient (open loop VBR) does not imply constant video quality encoding.

Video object streams were VBR-encoded with following parameters: lossless shape coding (alpha threshold = 0, changeCRdisable = 0), binary alpha channel, 8 bits/pixel, and the following options enabled: error resilience, data partitioning, reversible VLC, AC/DC prediction and SADCT coding. In our experiment, we selected a lossless shape because each video object was encoded independently. Lossy encoding of independently encoded objects belonging to the same original frame created shape artifacts at the composition layer due to misalignment of edges of objects.

5.5.1.1 Content Analyzer

In this experiment, content features were obtained directly from the MPEG-4 video encoder. Alternatively, they can be extracted from the encoded MPEG-4 video object streams in a way similar to the MPEG-2 experiment, described in Section 5.5.2. The following video object features were used: object size (in pixels), average and variance of motion vectors, and average energy of AC DCT coefficients. Our experiments have shown that the accuracy can be improved by including the encoder-related features in the feature vector. The following MPEG-4 encoder-specific features and parameters were used: quantization parameter q, frame type (I, P, B); number of bits for shape, motion, texture, and headers. These features can be extracted relatively easily from the compressed video stream at both server and network nodes in real time. Additionally, we have used PSNR that was computed directly by the MPEG-4 encoder.

5.5.1.2 Utility generator

A parameterized form of the utility function for each video object was obtained in the following way. The utility generator adopted an automated method for computation of subjective video quality. The method is based on modified version of the Picture Quality Assessment System, a model of human visual system introduced by Hamada in [73].

To support utility estimation corresponding to individual MPEG-4 video objects, the original Picture Quality Assessment System was modified in the following way: the system was modified such that it uses three CIF streams as an input: (i) an original video sequence, (ii) a decoded MPEG-4 video object sequence corresponding to an MPEG-4 video object stream encoded at given quantization coefficient and (iii) a binary video object mask sequence obtained from semi-automatic video

segmentation system [77]. The binary video object mask selects the region for which the quality score was computed. The output from the system is the mean opinion score (MOS) for MPEG-4 video object in a range from 1 (lowest) to 5 (highest) [74]. Similarly, utility values (i.e., quality) corresponding to different quantization parameters q are obtained by using the MPEG-4 encoded video object stream with different q. Quality estimations corresponding to video object streams obtained using different quantization coefficients served as sampling points for parameterized utility function.

In the experiment, utility function was defined by seven sampling points x_q corresponding to seven quantization parameters:

$$\mathcal{U} = \{x_q;\ q = 1, 5, 10, 15, 20, 25, 31\},\ \ x_q = \{r(q), u(q)\} \tag{5.1}$$

where $r(\cdot)$ denotes rate, $u(\cdot)$ utility, and q the quantization parameter.

5.5.1.3 Utility Clustering Modules

The utility clustering module clusters the "shape" of utility function of video objects into a set of utility classes. The "shape" was specified by the feature vector consisting of discrete values corresponding to sampling points of utility function (defined in Equation 5.1. In the following, two different methods based on two different utility classification models, composite and joint, are compared.

The composite model consists of two independent 9-class classifiers: rate-only and utility-only. According to utility function defined in Equation 5.1, rate-only classifier is used for classification of rate $r(q)$ and utility-only classifier is used for classification of utility $u(q)$; both $r(q)$ and $u(q)$ are indexed by quantization parameter q. Consequently, the rate-only model uses only a subset of parameters from

$\mathcal{U}$, namely its rate components $r(q)$ as a feature vector. Similarly, the utility-only model uses utility components $u(q)$ only. The classification results from both independent classifiers are combined to create 81-state composite model. For example, assume that the video object's rate $r(q)$ is classified into class a_r and its utility $u(q)$ is classified into class b_u. In that case, video object classification according to composite model is expressed as $\{a_r, b_u\}$.

The joint model uses all 14 parameters (i.e., $r(q_i)$ and $u(q_i)$) defining utility function $\mathcal{U}$ as a feature vector. Similar to the composite model, the joint model classifies utility function into 81 classes. There is a distinction and tradeoff between the use of composite and joint models. Given the same number of classes, the joint model can obtain clustering results more accurately, because it uses a full set of utility function parameters as the feature vector. Additionally, it can be directly used for estimation of the utility function for video streams under dynamic rate shaping adaptation (i.e., determining utility based on available bandwidth). On the other hand, the composite model is better suited for encoder controlled adaptation since both of its constituent models (rate only and utility only) are indexed in terms of quantization coefficient q.

Functions associated with the utility clustering module and decision-tree generator were simulated using publicly available machine learning tools. In particular, the utility clustering module was realized by Autoclass III [78] and the decision-tree generator was based on OC1 software [79]. Autoclass III is Bayesian unsupervised classifier that predicts class membership given unlabeled test cases. Autoclass III was configured to automatically select a fixed number of 9 classes for rate-only and utility-only models and 81 classes for the joint model. OC1 is a supervised machine learning system based on oblique decision trees. Decision trees of this form consist of linear combinations of the attributes at each internal node and can be viewed as

more general forms of axis-parallel univariate decision trees.

5.5.1.4 Results

In each one of the three experiments, we compare two utility clustering models: composite and joint models. In the first two experiments, we include two distinct models using I-frames and P-frames only. In particular, in the first experiment we cluster video objects using I-frames only and in the second experiment, we cluster video objects using P-frames only. In the third experiment, we apply the clustering models to include both I- and P-frames.

As we have already mentioned, a composite model consists of two intermediate utility-only and rate-only clustering models that are independent of each other. The utility-only model is depicted in Figure 5-5 The rate-only model is depicted in Figure 5-6. Each graph illustrates nine different classes of distinct shape. Each class is shown by its mean value, computed among all members of particular class. In our example, nine classes were obtained using the Autoclass III software. Note that each clustering model is based on different feature vectors. In particular, the feature vector of the utility-only model consists of seven utility values for each video object corresponding to different values of q. Similarly, rate-only model consists of seven rate values.

Both utility-only and rate-only models are combined to form the composite model. Figure 5-7 depicts resulting composite model consisting of 81 utility classes. The model is based on previously obtained rate-only and utility-only models. Each utility class is shown by its mean utility function $\mathcal{U}$. Figure 5-7 illustrates different shapes of utility function corresponding to different utility classes.

On the other hand, the joint model is formed directly from all 14 features of utility function $\mathcal{U} = \{x_q;\ q = 1, 5, 10, 15, 20, 25, 31\}$. For comparison, the joint

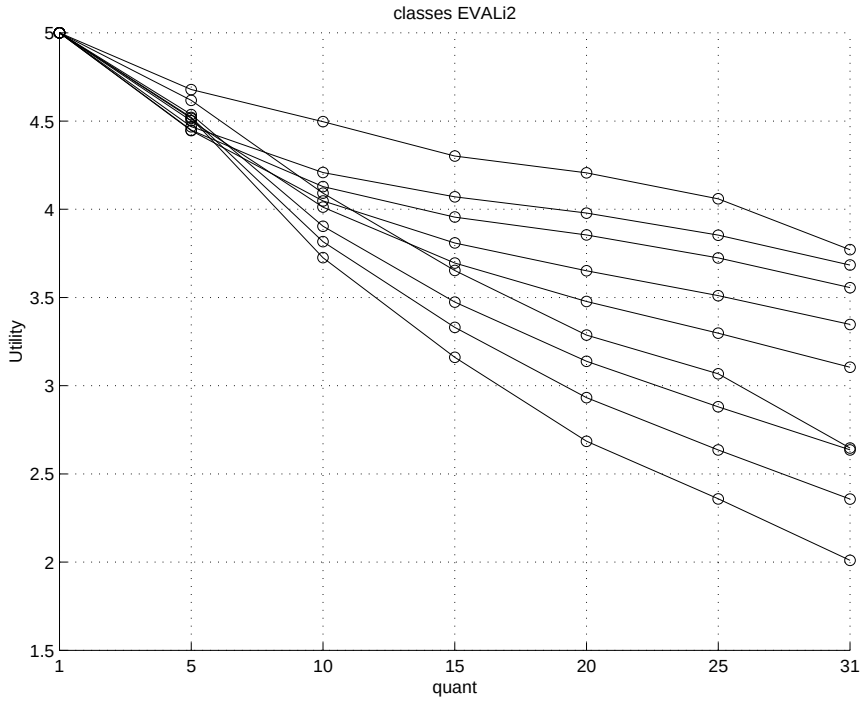

Figure 5-5: UT9 utility classification.

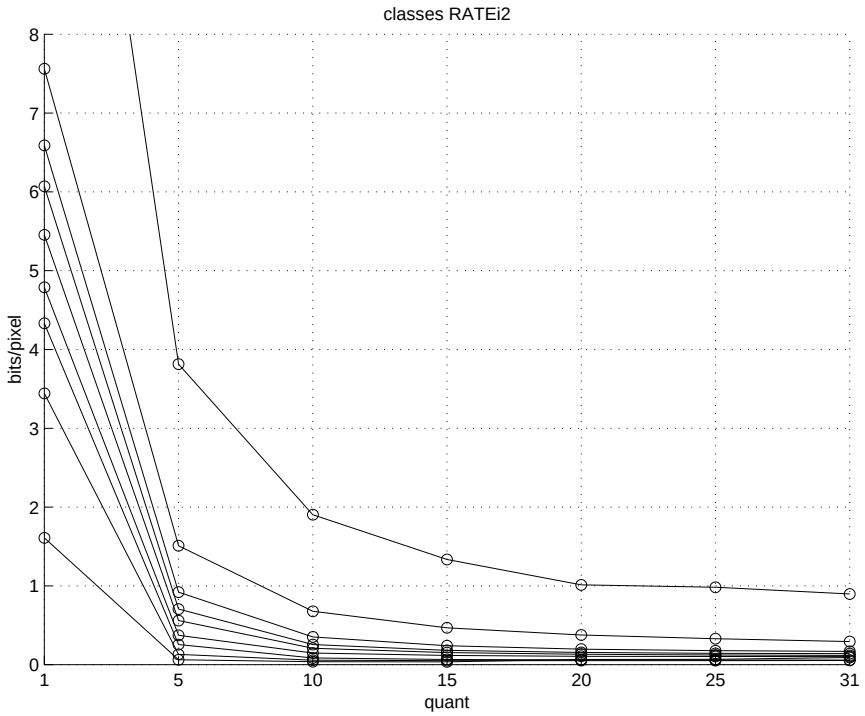

Figure 5-6: RT9 rate classification.

model is depicted in Figure 5-8. Similarly, each utility class is shown by its mean utility function $\mathcal{U}$.

In general, clusters obtained using composite and joint models are different. Because the composite model combines two independent models, some utility classes corresponding to particular rate-only and utility-only classes may not be populated (e.g., no video object is classified into that particular utility class). Consequently, a composite model may not represent optimal clustering. On the other hand, a joint model can obtain clusters that are not possible to capture by the orthogonally projected rate-only and utility-only models of a composite model. This feature leads to better clustering results compared to the composite model. Our results, summarized in Table 5.1, indicate that this is also the case in our experiments.

The results obtained by composite and joint utility clustering models are summarized in Table 5.1. In our experiments, the quality of clustering is measured by classification consistency and mean square error (MSE). Classification consistency of model $\mathcal{P}$ is defined as $\mathcal{F}\{\mathcal{P}\} = 10 \log S\{\mathcal{P}\}/G\{\mathcal{P}\}$, where $G(\mathcal{P})$ is degree of grouping and $S(\mathcal{P})$ is degree of separation [75]. In general, higher values of $\mathcal{F}$ are related to the model in which better clustering is obtained (e.g., small distances between members in the same class and large distances between classes). Mean square error (MSE) is measured using the distance between the actual utility function of each video object and characteristic utility function (e.g., mean utility function for each class) of the class into which the particular video object is classified. As expected, the better clustering, indicated by higher values of $\mathcal{F}$, corresponds to the joint model. Similarly, the better clustering, with the exception of the I-frame test case, is confirmed by lower MSE values for the joint model, compared to the composite model.

The results obtained by a decision-tree classifier are summarized in Table 5.2.

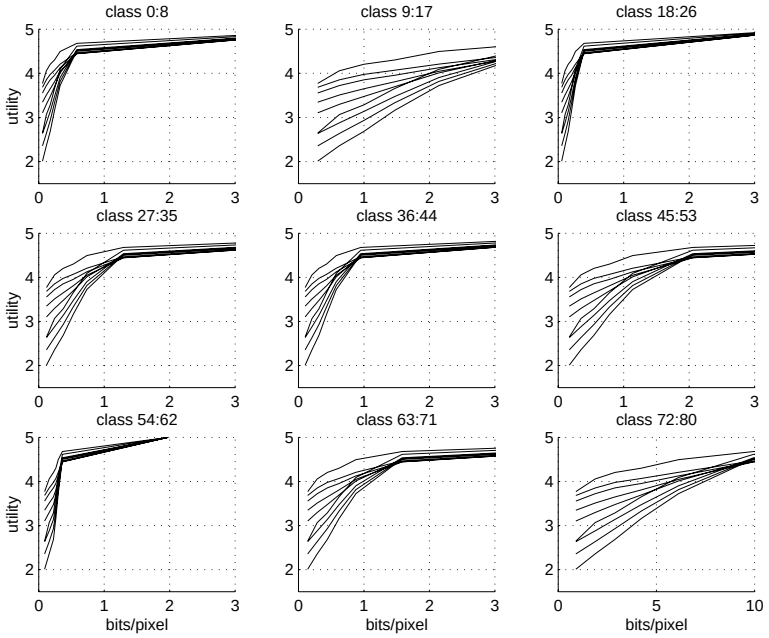

Figure 5-7: Composite utility clustering model for I-frames.

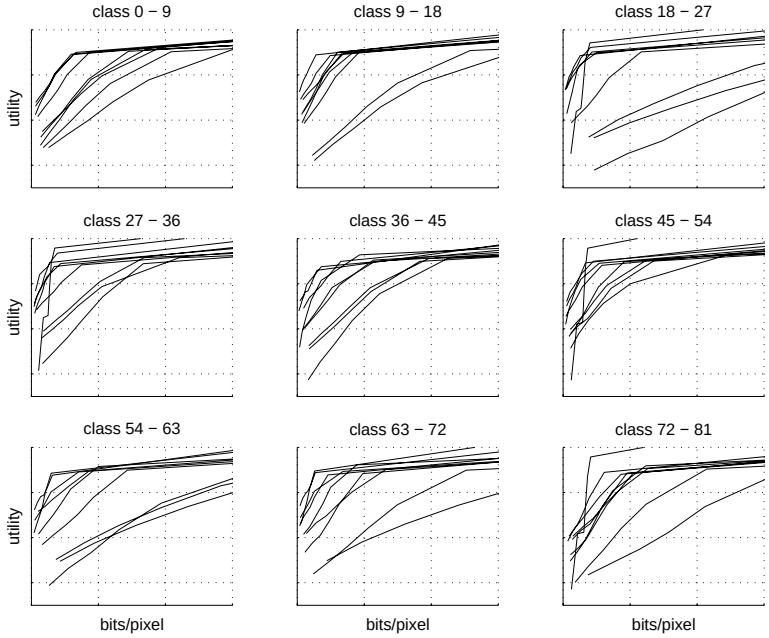

Figure 5-8: Joint utility clustering model for I-frames.

	I-frame		P-frame		I- & P-frame	
	comp	joint	comp	joint	comp	joint
$\mathcal{F}$	26.61	28.49	25.21	31.90	25.07	32.13
MSE	1.102	3.22	3.22	2.60	3.32	3.01

Table 5.1: Comparison of composite (comp) and joint utility clustering models.

Classification accuracy measures the percentage of video objects that were classified into correct classes. In our experiments, composite and joint decision-tree classifiers are constructed for each I-, P-, and I- & P-frame test case. They directly correspond to composite and joint utility clustering models in Table 5.1. The results were obtained using video objects that were not used to train the adaptive content classification loop (i.e., to cluster utility functions and generate decision-tree classifier). Because (i) the decision-tree classifier is based on the content feature vector and (ii) optimizations are performed in the decision-tree classifier to reduce complexity of decision tree, some of the video objects are not classified correctly into the utility class to which they belong. This fact is illustrated by classification accuracy and, with the exception of the joint model for I-frame, by a decrease in $\mathcal{F}$ and an increase of MSE, compared to results in Table 5.1.

Although classification accuracy is similar for both composite and joint models, it is somewhat higher for the composite model and I- and P-frame test cases. In contrast, with the exception of the I-frame case, joint models give better results in terms of MSE. In all cases, the joint model achieved better classification consistency $\mathcal{F}$ compared to the composite model.

Based on our results, it appears that while composite models perform slightly better for homogeneous test cases (i.e., for I-frames and P-frames test cases) in terms of classification accuracy, joint models perform better for both P-frame and combined I- & P-frame test cases in terms of both classification consistency and MSE. Consequently, models based on combination of simple classifiers (i.e., com-

	I-frame		P-frame		I- & P-frame	
	comp	joint	comp	joint	comp	joint
accuracy (%)	88.62	78.93	80.51	78.75	83.89	86.78
$\mathcal{F}$	25.84	30.25	24.43	30.93	24.54	31.74
MSE	1.57	4.07	4.26	3.77	3.58	3.31

Table 5.2: Comparison of composite (comp) and joint decision-tree classifiers.

posite models) may be sufficient to construct decision-tree classifiers that operate on video objects of the same type (i.e., I- or P-frames only). Models based on joint rate-utility feature vectors (i.e., joint models) are more appropriate in cases where the training pool contains video objects of different types (i.e., I- & P-frames combined).

5.5.2 MPEG-2 Content-based Utility Estimator

In this experiment, we have used MPEG-2 video stream consisting of 3000 frames created from the movie "Forrest Gump" using the Columbia University MPEG-2 software encoder. Our experiment was based on a subset of the trace, namely 734 frames consisting of P-frames for which content features were obtained by the automated content analyzer.

5.5.2.1 Implementation

The content analyzer operates as follows. First the content analyzed conducts video scene detection, then video object detection, and finally, content feature extraction. Because the analyzer operates in the compressed domain, the original MPEG-2 decoder was simplified to contain only parts necessary for scene detection, video object detection and content feature estimation. In particular, the computationally intensive inverse DCT transformation was omitted. This simplification results in real-time performance on a general-purpose workstation. For example, on SUN

SPARCstation 5, content analysis of 3000 frames completed in 16.5 s; that is in average 5.5 ms per frame (i.e., before the next frame needs to be processed).

The following content features are extracted directly from the encoded MPEG-2 video stream: current frame size, number of objects (max. 2), object size (in macroblocks), average and variance of motion vectors, number of forward-predicted macroblocks, number of DCT-encoded macroblocks, camera operation parameters (viz. translation, zoom and divergence speed), and average energy of AC DCT coefficients.

Contrary to the MPEG-4 experiment, where the video quality was evaluated for video objects, the video quality in this MPEG-2 experiment was evaluated for the entire frame. In addition, video objects in MPEG-4 experiment were segmented using manually-assisted technique [77], therefore segmentation was more accurate than in the approximate compressed-domain technique, used in MPEG-2 experiment. We have found that high classification accuracy of 80% to 85%, achieved in MPEG-4 experiments, cannot be attained with a fully automated MPEG-2 content analyzer (achieved accuracy was 55% to 65%). However, the accuracy can be greatly increased by including PSNR as one of the content features. Given that content-based utility function estimator is collocated with real-time video source, PSNR can be obtained directly from the encoder.

Similarly to MPEG-4 utility estimator, the utility clustering modules were based on Autoclass III software tool [78]. Autoclass III was configured to automatically select a fixed number of 17 classes used during classification. Likewise, the decision tree generator was based on OC1 software [79].

5.5.2.2 Results

Utility function for each frame was constructed from 21 rate sampling points. The samples were obtained using the dynamic rate shaping (DRS) system [83]. The sampling points were evenly distributed in the range from 0% to 100% of the frame bit-rate. Utility functions were classified at the utility clustering module. Figure 5-9 illustrates one snapshot of the classification results. In this example, 17 classes are formed out of a total 734 utility functions. Each sub-graph illustrates all the individual utility functions within a single class. These are shown as shaded curves in Figure 5-9. The single dark curve represents the characteristic utility function of that utility class, which was approximated as 10th percentile of the utility curves belonging to each class. This way, 90% of utility functions belonging to a particular utility class will not be overestimated (in terms of quality) by its characteristic utility function. In our experiment, we have chosen 10th percentile to correspond to the classification accuracy of utility selector (i.e., 91%).

The classes are numbered from 0 to 16. The number in parenthesis indicates the number of utility functions in each class. In this case, the number of utility functions in a single class ranges from 16 to 67. The figure shows the good classification performance of Autoclass III software, as the utility functions of similar shapes are clustered into the same utility class.

After utility classification, the decision tree generator forms the decision tree from the extracted content features serving as feature points in supervised classification. The decision tree represents a hierarchical mapping between content feature vectors and utility classes. Figure 5-10 illustrates the classification results based on the decision tree using the same representation as Figure 5-9.

The decision tree accuracy is crucial to the proper functioning of the system. The decision tree estimates the utility function given a set of content features that

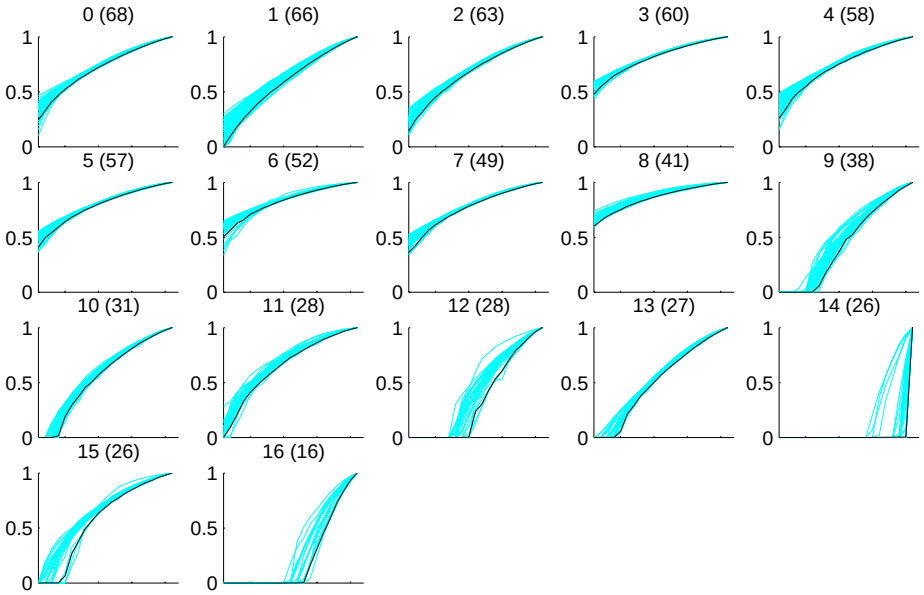

Figure 5-9: Utility classification.

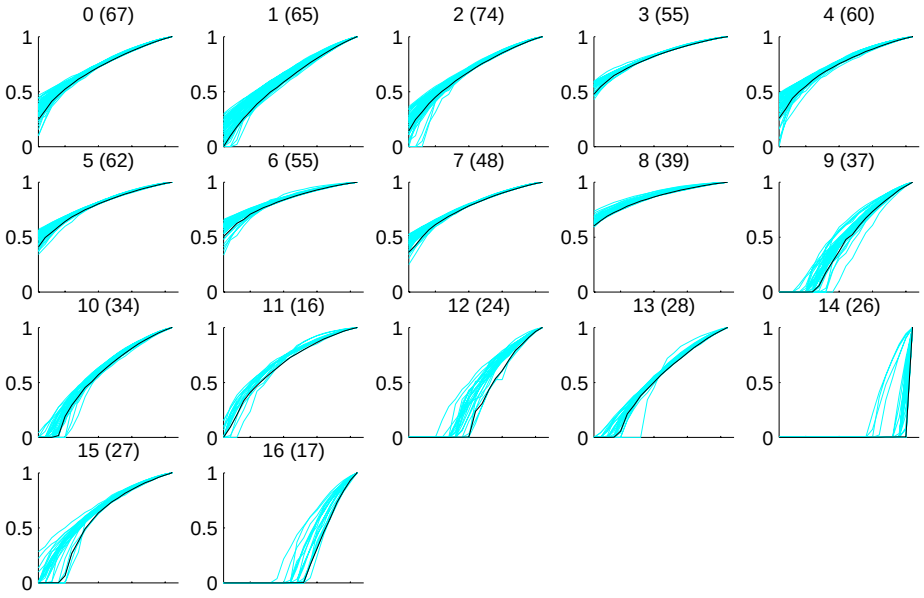

Figure 5-10: Content classification.

Utility Class	1	2	3	4	5	6	7	8	9
Accuracy (%)	100	100	93.88	90.00	93.10	100	100	98.08	92.65
Utility Class	10	11	12	13	14	15	16	17	
Accuracy (%)	93.65	92.59	92.42	90.32	81.58	53.57	76.92	78.57	

Table 5.3: Decision Tree Accuracy (MPEG-2 experiment).

are easy to obtain from compressed video streams. However, since content features do not contain direct information regarding utility functions, mismatches in classification between content features and utility classes can occur. In other words, at some instances, a decision tree is not able to correctly identify a utility class based on analyzed content features alone. This effect can be observed in Figure 5-10. For example, comparing classes 9 in Figures Figure 5-9 and Figure 5-10, one can find several utility functions that are incorrectly classified into class 9 by decision tree. In practice, this will lead to the utility selector selecting the wrong characteristic utility function.

Table 5.3 summarizes the accuracy of the decision tree for each of the 17 utility classes as obtained during the simulation experiments. The overall classification accuracy of the whole set of utility functions was found to be 91%; that is to say, among the total 734 video objects, 91% of them were classified correctly using content features. This high level of classification accuracy demonstrates the viability of the approach.

5.6 Conclusion

We presented a new framework for real-time utility function estimation. We demonstrated that the system for real-time utility function estimation could be constructed using a unique content-aware video classification approach and general machine learning algorithms. The utility functions for video objects can be used for media

scaling in future low bandwidth and wireless networks. Our results indicated the feasibility and relatively high accuracy of such a system.

Chapter 6

Key Issues and Conclusions

The content-aware framework is based on the recognition of a strong correlation among video content, required network resources (bandwidth), and the resulting video quality (utility function). The content-aware principle recognizes the importance of using content features in estimating the traffic or quality models. The content features are used as the bridge in linking the activities in visual content to bandwidth requirements.

In the content-aware framework, media content is extracted automatically and used to predict video quality under various manipulations (e.g., transcoding) and network resource requirements. When we refer to "media content", we mean the multimedia features that can be analyzed by the machine. Examples include visual features (e.g., motion, complexity, size, and spatio-temporal relationships) of the scenes or objects. These features can be systematically analyzed and are very likely to be present in future multimedia content representations, such as MPEG-4 [92] and MPEG-7 [93].

The content-aware principle can be used for many applications. In dynamic resource allocation (DRA), it can be used for real-time video traffic prediction. Alternatively, it can be used for utility function generation to facilitate the network-

wise scalability. It can be used in selecting the optimal transcoding architecture and content filtering in a pervasive computing environment. The media object scalability through the use of utility functions has been included in object description schemes for Universal Multimedia Access (UMA) of MPEG-7 [94].

In the following sections, we revisit and summarize some of the key issues and important design strategies during the research work of this thesis.

6.1 Issues in Video Content Analysis

Content features are extracted from video streams by the content analyzer. The estimation of content features can be done in either uncompressed or compressed domains. Processing in the compressed domain reduces computation because frames do not need to be converted back to the uncompressed (original) domain [59].

We have implemented a real-time content analyzer in the compressed domain that is based on fully automated methods of content analysis. Because the automatic analyzer operates directly in the compressed domain, the decoder was simplified to contain only the parts that are necessary for activity period detection, video object detection and content feature estimation. In particular, the computationally intensive DCT function was omitted. This simplification resulted in a real-time performance on a general-purpose workstation. For example, on a SUN SPARCstation 5, it was possible to analyze each video frame for its content in less then 10 ms.

6.2 Issues in Bandwidth Prediction

One example of the purpose of a content-aware framework is prediction of bandwidth requirements for live video. The traffic prediction module is a critical element in adaptive networking systems such as dynamic resource allocation. It predicts

resource requirements for the current activity period. In Chapter 3, we described an approach based on two assumptions. First, recognition of distinctive classes of activities can be achieved by content feature analysis. Second, members of each activity class have consistent traffic models.

However, in practical applications, content features are extracted by automatic processes, which may cause some errors. In addition, videos of different activity types may produce similar traffic traces. The goal is to predict the traffic based on automatically extracted features. Distinction of activity types is not needed explicitly. Therefore, in an automated system, each activity period is directly mapped to a traffic class, without categorizing its activity type.

In our experiments, publicly available machine learning tools were used to simulate the above-described content-based classifier. The time spent on the clustering operation is not critical because it is performed during off-line training or on-line adaptation, and does not affect real-time performance.

The accuracy of the content-based classifier is important. Misclassification occurs when an activity period is not mapped to the traffic class with the most accurate traffic descriptor. We have simulated the generation of parameters of the content-based classifier using one half of the activity periods identified in the video and verified its accuracy on the remaining subset. In our experiments, a high classification accuracy of 86.14 % was achieved.

Performance study of content-based DRA was based on trace-driven simulations. A trace-driven simulator was developed for that purpose. Results were obtained using a single 54000-frame-long trace (30 minutes) of an MPEG-2 encoded movie.

Network simulations revealed that both the content-based approaches achieve better performance (in terms of link utilization) than other existing schemes (RVBR). The link utilization achieved by RVBR was substantially less than the utilization

achieved using the "APD auto" scheme (about 55% - 70% difference). In addition, CBRC achieved mostly better utilization than existing schemes.

The superior performance of the content-based dynamic resource allocation, based on the content-aware framework, can be attributed to its two distinguishing features. First, the fact that it is able to track changes in visual content (and therefore changes in bit-rate); this is accomplished mainly by detecting discontinuities in visual content. Second, content-aware models facilitate the use of effective content classification methods which improve resources prediction accuracy. This is in contrast to traditional prediction schemes that use only bit rate and network buffer occupancy in their heuristics segmentation and resource prediction algorithms.

6.3 Utility Function Estimation

Utility functions represent a powerful framework for characterizing the ability of applications to adapt to varying network conditions. Specifically, in the context of bandwidth allocation, utility functions indicate a media object's quality as a function of available bandwidth.

In practice, the estimation of utility functions requires repetitive computation of quality metrics with different encoder parameters such as quantization or dynamic rate shaping (DRS) parameters. The process of repetitive estimation is the main cause of the large amount of calculations required for utility function evaluation. To the best of our knowledge, a system that allows efficient estimation of subjective utility function in real time does not exist. We demonstrated a new system for the speedup of generation of utility function based on the content-based classification technique that allows estimation of utility in real-time [97].

The acceleration technique does not explicitly compute utility functions for each video object. Rather, the content-aware principle is applied and machine learning

techniques are used. The system uses video content, represented by a limited set of content features, to determine the utility class of an object. Because video content can be dynamically extracted from compressed video streams, this technique is suitable for real-time applications.

Accuracy of the MPEG-2 and MPEG-4 content-based utility estimator was evaluated in [97]. For the experiment using MPEG-2 video (17 utility classes), the classification accuracy of the whole set of utility functions was 91%; that is to say, among the total 734 video frames, 91% of them were classified correctly using content features. Similarly, a high classification accuracy of 80% - 85% was achieved using MPEG-4 traces.

6.4 Content-aware Network Architecture

The content-based video communication framework presented in this thesis is likely to have implications related to the architecture of future multimedia networks. The content-aware networking differs from traditional networks. First, it takes into account a number of factors influencing the quality, as perceived by the end-user: video content, encoding technique, network characteristics, terminal capabilities and user preferences. Second, the architecture comprises several novel features enabling the support for application-specific adaptation needs inside networks. The core of the system is the utility generator formulating utility functions in real-time. By exploiting video content and user preference, the utility functions and scaling profiles capture application-specific adaptation capability and preference, respectively.

The results presented in previous chapters demonstrated that the combination of user-preferred and application-specific adaptation and media scaling inside the network is a promising approach for delivery of highly scalable multimedia materials over the time-varying networks.

6.5 MPEG-7 Content Description for Universal Multimedia Access

Multimedia Content Description Interface (MPEG-7) is an ongoing standardization work started in 1999 by Moving Picture Experts Group (MPEG). One of the goals of the MPEG-7 standard is to allow a generic description of the audio-visual material in terms of it's content, purpose, ownership rights, audience, network resource requirements, playback capabilities, etc. The media description can be used in database applications for fast and efficient searching and filtering, media composition and transmission. The MPEG-7 will allow applications to have seamless access to "image/video/audio" databases based on standardized media description language. Besides database applications, the MPEG-7 can also be used for content description of interactive or real-time media services such as TV, video-on-demand, radio stations, etc.

A great part of the MPEG-7 work is still in a preliminary stage. In essence, the standard will describe a set of MPEG-7 descriptors (D) and description schemes (DS). The MPEG-7 systems layer will deal with delivery issues. One of the applications of MPEG-7, called "Universal Multimedia Access" (UMA) was recently added to the MPEG-7 Application document [106]. The primary goal of UMA is the description and management of alternative versions of the audio-visual document in terms of their resource requirements and flexibility toward transcoding. Different abstraction levels of the audio-visual material will facilitate flexible adaptation to the network conditions and receiver capabilities [107, 22].

MPEG-7 will likely standardize scaling operations that are well suited for dynamic scaling of the multimedia objects during their transport over the bandwidth-limited networks. The use of media scaling can substantially increase the perceptual

quality of the multimedia presentation in the case of temporal bandwidth variation or congestion in wireless networks or Internet. The need for media scaling operations at the network comes from the time-scale mismatch between the content (slow time-scale) and network bandwidth variation (fast time-scale). The scaling profile indicates the sequence of preferred scaling operations on multimedia objects and their corresponding utility functions. The scaling profile can be used in networks for "utility-fair" scaling during temporal bandwidth limitations. The scaling profile should contain a preferred set of scaling operations at the network, bandwidth utility function, and the current scaling operation and utility operating point.

Multimedia content description for UMA will add a number of descriptors into the base MPEG-7 standard. This information will be used by terminal devices when deciding which specific variation of the document to retrieve. The decision will be based on the device's communication availability, processing, power and presentation capabilities. UMA content description information will also be used by transcoding engines to create the requested version of the document in real-time. Alternatively, content descriptors will likely be used by content-based media transcoding agents at intelligent network switches or wireless base stations for resource management. This will allow for possible gradual and non-destructive audio-visual quality degradation in the case of temporal bandwidth variations.

In relation to UMA, the multimedia data can be described by the following attributes: purpose, resource, priority, value, variation, utility function, traffic descriptors, scaling profile, variation hints, and transcoding hints.

The *purpose* attribute describes the role of the multimedia object. For example, multimedia data purpose can take the following values: advertisement, decoration, navigation, logo, etc. [108].

The *resource* attribute describes the requirements of the multimedia data for

delivery, processing and rendering. For example, resource information can indicate that a particular video requires a minimum streaming bandwidth, decoder buffer size. In addition it may indicate a required client device vendor, model, class of device (phone, PDA, printer, etc.), screen size, colors, available bandwidth, CPU, memory, input device, secondary storage.

The *priority* information describes the importance of a multimedia object in a presentation. The object priority may aid the process of adaptation and transcoding. For example, if the priority of the object is low, it may be scaled, translated, summarized, removed or replaced if necessary. If possible, in the case of adaptation multimedia, objects of high priority will be preserved.

The *value* information describes the value of the information contained within the multimedia data [109]. The value may depend on the interest of the user, or may be determined by the content author or publisher.

The *variation* information describes current variation of the multimedia item. In general, different variations of the content can be described as: translation, summarization, extraction, substitution, visualization, and scaling. For example multimedia object B can be a variation of another object A. The fidelity attribute of the variation relationship indicates the subjective or objective quality representing its usability for the presentation.

The *utility function* is specified in relation to the currently selected variation. The function can be defined as continuous (polynomial, exponential, etc.) or defined on a discrete set of points or categories. The utility can take values 1 to 5 or distortion scale as recommended by ITU [74]. In networks, utility functions can be used for utility-based bandwidth allocation.

The *traffic descriptors* provide information about required bandwidth needed for real-time streaming of the media. Several traffic models can be supported for

various networks.

The *variation hint* information is an ordered set of operations provided by the content publisher to suggest what should be done to a particular multimedia item within a multimedia presentation.

Besides the description of variation of a multimedia document, Universal Multimedia Access requires some attributes and description tools to aid the scaling operations. These attributes, grouped in transcoding hints, will be storage-format dependent.

The main objective of Universal Multimedia Access is to enable adaptive transport and delivery of multimedia to various client devices with limited communication, processing, storage and display capabilities. As part of MPEG-7, UMA will represent an important application aim to be an integral part of future multimedia information services. Our work of utility function generation (Chapter 5) and resource requirement estimation (Chapter 3) can be used to generate descriptors needed in UMA.

References

[1] L. Chiariglione, "MPEG and Multimedia Communications", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 7, No. 1, February 1997.

[2] J. R. Nicol, Y. S. Gutfreund, J. Paschetto, K. S. Rush, and Ch. Martin, "How the Internet Helps Build Collaborative Multimedia Applications", *Communications of the ACM*, Vol. 42, No. 1, January 1999.

[3] R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin, "Resource ReSerVation Protocol (RSVP), Version 1 Functional Specification", *IETF Internet RFC2205*, September 1997.

[4] J. Wroclawski, "The Use of RSVP with IETF Integrated Services", *IETF Internet RFC2210*, September 1997.

[5] D. Black, S. Blake, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Services", Internet RFC 2475", *IETF Internet RFC2475*, December 1998.

[6] S. Butler and A. P. Parkes, "Filmic space-time diagrams for video structure representation", *Signal Processing: Image Communication*, No. 8, 1996.

[7] Cheng-Shang Chang, and Joy A. Thomas, "Effective Bandwidth in High-Speed Digital Networks", *IEEE Journal on Selected Areas in Communications*, Vol. 13, No. 6, August 1995.

[8] G. de Veciana, G. Kesidis, and J. Walrand, "Resource Management in Wide-Area ATM Networks Using Effective Bandwidths", *IEEE Journal on Selected Areas in Communications*, Vol. 13, No. 6, August 1995.

[9] P. Pancha and M. E. Zarki, "MPEG Coding for Variable Bit Rate Video Transmission", *IEEE Communications Magazine*, May 1994.

[10] J. Beran, R. Sherman, M. S. Taqqu, and W. Willinger, "Long-Range Dependence in Variable-Bit-Rate Video Traffic", *IEEE Transactions on Communications*, Vol. 4 3, No. 2/3/4, February/March/April 1995.

[11] A. Baiocci, N. B. Melazzi, M. Listanti, A. Roveri, and R. Winkler, "Loss Performance Analysis of an ATM Multiplexer Loaded with High-Speed ON-OFF Sources", *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 3, April 1991.

[12] J. Bergen, P. Anandan, K. Hanna, and R. Hingorani, "Hierarchical Model-Based Motion Estimation", *Proceedings of the 2nd ECCV*, 1992, pp. 237-252.

[13] S. Gumbrich, H. Emgrunt, and T. Brown, "Dynamic bandwidth Allocation for Stored VBR Video in ATM End Systems", *Proceedings of IFIP'97.*

[14] ATM Forum Technical Committee, "Traffic Management Specification Version 4.0", AFTM 0056.0000, April, 1996.

[15] Hui Zhang and Domenico Ferrari, "Improving Utilization for Deterministic Service In Multimedia Communication", *Proceedings of IEEE International Conference on Multimedia Computing and Systems*, 1994.

[16] A. T. Campbell, G. Coulson, and D. Hutchison, "Transporting QoS Adaptive Flows", *ACM Multimedia Systems Journal, Special Issue on QoS Architecture*, Vol. 6, No. 3, 1998.

[17] T. Sikora, "The MPEG-4 Video Standard Verification Model", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 7, No. 1, February 1997.

[18] N. Matoba, Y. Kondo, M. Yamashina, and T. Tanaka, "Characteristics of Video Communication System in Mobile Radio Channel", *IEICE Transactions on Communications*, Vol. E80-B, No. 8, August 1997.

[19] A. A. Webster, C. T. Jones, M. H. Pinson, S. D. Voran, S. Wolf, "An objective video quality assessment system based on human perception", *SPIE*, Vol. 1913/15, 1993.

[20] H. Ji, "An Economic Model for Bandwidth Allocation in Broadband Communication networks", *Proceedings of ICC*, 1996.

[21] H. Jiang and S. Jordan, "The Role of Proce in the Connection Establishment Process", *European Transactions on Telecommunications*, Vol. 6, No. 4, July-August 1995, pp. 421-429.

[22] R. Mohan, J. R. Smith, and Ch.-S. Li, "Adapting Multimedia Internet Content for Universal Access", *IEICE Transactions on Multimedia*, Vol. 1, No. 1, March 1999.

[23] H. Yamanoto, "Digitalization of Mobile Communication Systems", *IEICE Transactions on Communications*, Vol. E80-B, No. 8, August 1997.

[24] D. J. Reininger, D. Raychaudhuri, and J. Y. Hui, "Bandwidth Renegotiation for VBR Video Over ATM Networks", *IEEE Journal on Selected Areas in Communications*, Vol. 14, No. 6, August 1996.

[25] S. Chong, S.-q. Li, and J. Chosh, "Predictive Dynamic Bandwidth Allocation for Efficient Transport of Real-Time VBR Video over ATM", *IEEE Journal on Selected Areas in Communications*, Vol. 13, No. 1, January 1995.

[26] C. L. Williamson, "Dynamic Bandwidth Allocation Using Loss-Load Curves", *IEEE/ACM Transactions on Networking*, Vol. 4, No. 6, December 1996.

[27] H. Saito, "Dynamic Resource Allocation in ATM Networks", *IEEE Communications Magazine*, May 1997, pp. 146-153.

[28] S. Jordan, "Connection Establishment in High-Speed networks", *IEEE Journal on Selected Areas in Communications*, Vol. 13, No. 7, September 1995.

[29] A. Simonian, and J. Guibert, "Large Deviations Approximation for Fluid Queues Fed by a Large Number of On/Off Sources", *IEEE Journal on Selected Areas of Communications*, Vol. 13, No. 6, pp. 1017-1027, August 1995.

[30] V. S. Frost and B. Melamed, "Traffic Modeling For Telecommunications Networks", *IEEE Communications Magazine*, March 1994, pp. 70-81.

[31] D. Reininger, B. Melamed, and D. Raychaudhuri, "variable Bit Rate MPEG Video: Characteristics, Modeling and Multiplexing", *Proceedings of the 14th International Telegraffic Congress - ITC 14*, Antibes Juan-les-Pins, France, 6-10 June, 1994.

[32] B. Maglaris, D. Anastassiou, P. Sen, G. Karlsson, and J. D. Robbins, "Performance Models of Statistical Multiplexing in Packet Video Communications", *IEEE Transactions on Communications*, Vol. 36, No. 7, pp. 834-844, July 1988.

[33] P. Sen, B. Maglaris, N.-E. Rikli, and D. Anastassiou, "Models for Packet Switching of Variable-Bit-Rate Video Sources", *IEEE Journal on Selected Areas of Communications*, Vol. 7, No. 5, pp. 865-869, June 1989.

[34] P. Skelly and M. Schwartz, "A Histogram-Based Model for Video Traffic Behavior in an ATM Multiplexer ", *IEEE/ACM Transactions on Networking*, Vol. 1, No. 4, pp. 446-459, August 1993.

[35] G. Ramamurthy and S. Sengupta, "Modeling and Analysis of a Variable Bit Rate Video Multiplexer", *Proceedings of INFOCOM'92*, pp. 817-827, 1992.

[36] F. Yegenoglu, B. Jabbari and Y.-Q. Zhang, "Modeling of Motion Classified VBR Video Codecs", *Proceedings of INFOCOM'92*, pp. 105-109, 1992.

[37] A. Lazar, G. Pacifici and D. E. Pendarakis, "Modeling video sources for real-time scheduling", *Multimedia Systems*, pp. 253-266, 1994.

[38] J.-P. Leduc, P. Delogne, "Statistics for variable bit-rate digital television sources", *Signal Processing: Image communication*, No. 8, 1996.

[39] D. P. Heyman, T. V. Lakshman and A. Tabatabai, "Statistical Analysis and Simulation of MPEG-2 Coded Variable Bit Rate Video Traffic", *Proceedings of International Symposium on Multimedia Communications and Video Coding*, pp. 78-80, October 1995.

[40] D. P. Heyman and T. V. Lakshman, "Source Models for VBR Broadcast-Video Traffic", *IEEE/ACM Transactions on Networking*, Vol. 4, No. 1, pp. 40-48, February 1996.

[41] M. R. Frater, J. F. Arnold and P. Tan, "A New Statistical Model for Traffic Generated by VBR Coder for Television on the Broadband ISDN", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 4, No. 6, pp. 521-526, December 1994.

[42] R. M. Rodriguez-Dagnino, M. R. K. Khanari and A. Leon-Gracia, "Prediction of Bit-Rate Sequences of Encoded Video Signals", *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 3, pp. 305-314, April 1991.

[43] *Forrest Gump*, TM & Copyright © 1994 by Paramount Pictures.

[44] W.-ch. Feng, F. Jahanian, S. Sechrest, "An Optimal Bandwidth Allocation Strategy for the Delivery of Compressed Prerecorded Video", *CSE-Technical Report 260-95*, University of Michigan, August 1995.

[45] S. Lam and G. Xie, "Burst Scheduling: Architecture and Algorithm for Switching Packet Video", *Proceedings of IEEE INFOCOMM'95*

[46] G. Xie and S. Lam, "Real-time Block Transfer Under a Link Sharing Hierarchy", *IEEE/ACM Transactions on Networking*, 6(1), pp. 30-41, February 1998.

[47] ITU-T Rec. I.371, "Traffic Control and Congestion Control in B-ISDN", Petrh, U.K., November 6-14, 1995.

[48] P. Pan and H. Schulzrine, "YESSIR: A Simple Reservation Mechanism for the Internet", , August 1, 1997.

[49] W. Almesberger, T. Ferrari, J.-Y. Le Boudec, "SRP: a Scalable Resource Reservation Protocol for the Internet", , March 2, 1998.

[50] D. M. Lucantoni, M. F. Neuts, and A. R. Reibman, "Methods for Performance Evaluation of VBR Video Traffic Models", *IEEE/ACM Transactions on Networking*, Vol. 2, No. 2, pp. 176-180, February 1996.

[51] J. Y. Hui, "Resource Allocation for Broadband Networks", *IEEE Journal on Selected Areas in Communications*, Vol. 6, No. 9, December 1988, pp. 1598-1608.

[52] A. Adas, "Supporting Real Time VBR Video Using Dynamic Reservation Based on Linear Prediction", *Proceedings of IEEE INFOCOMM'96*, pp. 1476-1483.

[53] M. Grossglauser, S. Keshav, and D. Tse, "RCBR: A Simple and Efficient Service for Multiple Time-Scale Traffic", *Proceedings of SIGCOMM'95*, September 1995, pp. 219-230.

[54] H. Zhang and E. W. Knightly, "A new approach to support VBR video in packet-switching networks", *Proceedings of NOSSDAV'95*, April 1995, pp. 275-286.

[55] D. N. Tse, R. G. Gallager, and J. N. Tsitsiklis, "Statistical Multiplexing of Multiple Time-Scale Markov Streams", *IEEE Journal on Selected Areas in Communications*, Vol. 13, No. 6, August 1995, pp. 1028-1038.

[56] E. W. Knightly and H. Zhang, "D-BIND: An Accurate Traffic Model for Providing QoS Guarantees to VBR Traffic", *IEEE/ACM Transactions on Networking*, Vol. 5, No. 2, April 1997, pp. 219-231.

[57] P. Bocheck and S.-F. Chang, "A Content Based Video Traffic Model Using Camera Operations", *Proceedings of ICIP'96*, September 1996.

[58] H. Sanderson and G. Crebbin, "Image segmentation for compression of images and image sequences", *IEE Proceedings: Visual Image Signal Processing*, Vol. 142, No. 1, February 1995.

[59] J. Meng and S.-F. Chang, "Tools for Compressed-Domain Video Indexing and Editing", *Proceedings of SPIE Conference on Storage and Retrieval for Image and Video Database*, Vol. 2670, San Jose, February 1996.

[60] S. Siggelkow, R.-R. Grigat, A. Ibenthal, "Segmentation of Image Sequences for Object Oriented Coding", *Proceedings of ICIP'96*, September 1996.

[61] A. N. Netravali and B. G. Haskell, "Digital Pictures: Representation and Compression", Plenum Press, New York.

[62] L. Torres and M. Kunt, eds. "Video Coding: The 2nd Generation Approach", Kluwer Academic, Boston, Massachusetts, 1996.

[63] A. Murat Tekalp, "Digital Video Processing", Prentice Hall, 1995.

[64] S.-F. Chang, W. Chen, H.J. Meng, H. Sundaram, and D. Zhong, "VideoQ- An Automatic Content-Based Video Search System Using Visual Cues", *ACM Multimedia Conference*, Nov. 1997, Seattle, WA, also Columbia University/CTR Technical Report, CTR-TR #478-97-12. (demo: http://www.ctr.columbia.edu/videoq).

[65] J. Meng and S.-F. Chang, "CVEPS: A Compressed Video Editing and Parsing System", *ACM Multimedia Conference*, Boston, MA, Nov. 1996.

[66] H.-M. Hang and J.-J. Chen, "Source Model for Transform Video Coder and Its Application- Part I: Fundamental Theory", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 7, No. 2, April 1997.

[67] P. Bocheck and S.-F. Chang, "Content Based Video Traffic Modeling and its Application to Dynamic Network Resource Allocation", Columbia University CTR Report 486-98-20, New York, 1997.

[68] C. Aurrecoechea and A. T. Campbell, "A Survey of QoS Architectures", *Proceedings of 4th IFIP International Workshop on Quality of Service*, Paris, March, 1996.

[69] E. Gelenbe, X. Mang, and R. Onvural, "Bandwidth Allocation an Call Admission Control in High-Speed Networks", *IEEE Communications Magazine*, May 1996.

[70] K. L. Calvert, S. Bhattacharjee and J. Sterbenz, "Directions in Active Networks", *IEEE Communications Magazine*, October 1998, pp. 72-78.

[71] D. Reininger and R. Izmailov, "Soft Quality of Service with VBR$^+$ Video", *Proceedings of International Workshop on Audio-Visual Services over Packet Networks (AVSPN'97)*, Aberdeen, Scotland, UK, September. 15-16, 1997, pp. 207-211.

[72] D. Reininger, D. Raychaudhuri, and M. Ott , "A Dynamic Quality of Service Framework for Video in Broadband Networks", *IEEE Network*, November/December 1998.

[73] T. Hamada, S. Miyaji, and S. Matsumoto, "Picture Quality Assessment System by Three-layered Bottom-up Noise Weighting Considering Human Visual Perception", *Proceedings of SMPTE*, New York, 1997.

[74] Recommendation ITU BT.500-7, "Methodology for the Subjective Assessment of the Quality of Television Pictures".

[75] P. Bocheck and S.-F. Chang, "Content Based Dynamic Resource Allocation for VBR Video in Bandwidth Limited Networks", *Proceedings of the Sixth IEEE/IFIP International Workshop on Quality of Service (IWQoS'98)*, Napa, California, May 18-20, 1998.

[76] G. Bianchi, A. T. Campbell and R. R.-F. Liao, "Fair Support of Adaptive QoS Services over Wireless Networks", *Proceedings of the Sixth IEEE/IFIP International Workshop on Quality of Service (IWQoS'98)*, Napa, California, May 18-20, 1998.

[77] Di Zhong and Shih-Fu Chang, "AMOS: An Active System for MPEG-4 Video Object Segmentation", *International Conference on Image Processing 98*, October 4-7, 1998, Chicago, Illinois, USA.

[78] R. Hanson, J. Stutz, and P. Cheeseman, "Bayesian Classification Theory", *NASA Technical Report FIA-90-12-7-01.*

[79] S. K. Murthy, S. Kasif, and S. Salzberg, "A System for Induction of Oblique Decision Trees", *Journal of Artificial Intelligence Research*, 1994.

[80] "MPEG-4 VM software, MoMuSys-VFCD-V01-980507", *European ACTS project MoMuSys*, 1998.

[81] R. Liao and A. Campbell, "On Programmable Universal Mobile Channel", *Proceedings of ACM Mobicom'98*, Dallas, TX, October, 1998.

[82] R. Liao, P. Bouklee, and A. Campbell, "Online Generation of Bandwidth utility function for Digital Video", *Proceedings of PacketVideo'99*, New York City, Apr. 26-27, 1999.

[83] A. Eleftheriadis and D. Anastassiou, "Dynamic Rate Shaping of Compressed Digital Video", *Proceedings of 2nd IEEE International Conference on Image Processing (ICIP95)*, Arlington, VA, Oct. 1995.

[84] N. Yeadon, F. Garcia, D. Hutchison, and D. Shepherd, "Filters: QoS Support Mechanisms for Multipeer Communications", *IEEE Journal on Selected Areas in Communications, Special Issue on Distributed Multimedia Systems and Technology*, Vol. 14, No. 7, Sept. 1996, pp. 1245-1262.

[85] K. Lee, "Adaptive Network Support for Mobile Multimedia", *Proceeding of ACM Mobicom'95*, Berkeley, CA, Nov. 1995.

[86] A. Eleftheriadis, "Dynamic Rate Shaping of Compressed Digital Video", *Doctoral Dissertation*, Graduate School of Arts and Sciences, Columbia University, June 1995.

[87] J. Zamora, "Video-on-Demand Systems and Broadband Networks: Quality of Service Issues", *Doctoral Dissertation*, Graduate School of Arts and Sciences, Columbia University, 1998.

[88] A. Ortega and M. Khansari, "Rate Control for Video Coding over Variable Bit Rate Channels with Applications to Wireless Transmission", *Proceedings IEEE ICIP*, October 1995.

[89] G. de los Reyes, A. R. Reibman, J. C.-I. Chuang, and S.-F. Chang, "Video Transcoding for Resilience in Wireless Channels", *IEEE International Conference on Image Processing (ICIP'98)*, October 1998.

[90] ISO/IEC 11172-2:1993, "Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s – Part 2: Video", 1993.

[91] ISO/IEC 13818-2:1996, "Generic coding of moving pictures and associated audio information: Video", 1996.

[92] ISO/IEC 14496-2 CD, "MPEG-4 Visual", October 1997.

[93] ISO/IEC JTC1/SC29/WG11, "MPEG-7 Requirements Document, Coding of Moving Pictures and Audio", Vancouver, July 1999.

[94] J. R. Smith, Ch.-S. Li, A. Puri, Ch. Christopoulos, A. B. Benitez, P. Bocheck, and S.-F. Chang "Content Description for Universal Multimedia Access", *International Organisation for Standardisation ISO/IEC JTC1/SC29/WG11 MPEG99*, Vancouver, BC, July 1999.

[95] P. Bocheck, Yasuyuki Nakajima, and S.-F. Chang "Real-time Estimation of Subjective Utility Functions for MPEG-4 Video Object", *Proceedings of the Packet Video'99 (PV'99)*, New York, USA, April 26-27, 1999.

[96] P. Bocheck and S.-F. Chang "Content-based Video Traffic Modeling and its Application to Dynamic Resource Allocation", *Submitted to ACM/IEEE Transactions on Networking*, 1999.

[97] R. Liao, P. Bocheck, A. Campbell, and S.-F. Chang "Content-aware Network Adaptation for MPEG-4", *Proceedings of NOSSDAV'99*, June 1999.

[98] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: a transport protocol for real-time applications", *RFC1889, Internet Engineering Task Force*, January 1996.

[99] M. Andrews, S. C. Borst, F. Dominique, P. Jelenkovic, K. Kumaran, K. G. Ramakrishnan, and P. A. Whiting, "Dynamic Bandwidth Allocation Algorithms for High-Speed Data Wireless Networks", *Bell Labs Technical Journal*, July-September 1998.

[100] P. White, "RSVP and Integrated Services in the Internet: A Tutorial", *IEEE Communications Magazine*, May 1997.

[101] J. Ni, T. Yang, and D. H. K. Tsang, "Source modeling, queuing analysis, and bandwidth allocation for VBR MPEG-2 video traffic in ATM networks", *IEE Proceedings on Communications*, Vol. 143, No. 4, August 1996.

[102] R. Grunenfelder, J. P. Cosmas, S. Manthorpe, and A. Odinma-Okafor1, "Characterization of Video Codecs as Autoregressive Moving Average Precesses and Related Queuing System Performance", *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 3, April 1991.

[103] A. W. Bragg and Wushow Chou, "Analytic Models and Characteristics of Video Traffic in High Speed Networks", *Proceedings of MASCOTS '94, International Workshop on Modeling, Analysis and Simulation of Computer And Telecommunications Systems*, February 1994, Durham, NC.

[104] L. Wu, J. Benois-Pineau, Ph. Delagnes, and D. Barba, "Spatio-temporal segmentation of image sequences for object-oriented low bit-rate image coding", *Signal Processing: Image Communication*, No. 8, 1996, pp. 513-543.

[105] ISO/IEC 14496-1, "Information Technology - Coding of Audio-visual Objects, Part 1: Systems", *ISO/IEC JT1/SC 29/WG 11 Draft International Standard*, Dec. 1998.

[106] "MPEG 7 Applications document", *ISO/IEC JTC1/SC29/WG11/*, MPEG99, Seoul (Korea).

[107] "MPEG 7 Requirement document", *ISO/IEC JTC1/SC29/WG11/m2727*, MPEG99, Seoul (Korea).

[108] S. Paek and J. R. Smith, "Detecting Image Purpose in World-Wide Web Documents", *SPIE/IS&T Photonics West, Document Recognition*, January, 1998.

[109] R. Smith, R. Mohan, and C-S. Li. "Scalable Multimedia Delivery for Pervasive Computing", *ACM Multimedia*, Orlando, FL, November 1999.