An Architecture for Managing Virtual Circuit and Virtual Path Services on ATM Networks

Nikolaos Anerousis

Aurel A. Lazar

AT&T Laboratories
600 Mountain Avenue
P.O. Box 636
Murray Hill, NJ 07974-0636
tel. (908) 582-5499
e-mail: nikos@research.att.com

Department of Electrical Engineering
and
Center for Telecommunications Research
Columbia University
New York, NY 10027
tel. (212) 854-1747
e-mail: aurel@ctr.columbia.edu

Abstract

ATM networks rely on the Virtual Circuit (VC) and Virtual Path (VP) concept to provide unicast connection-oriented services with Quality of Service guarantees. This paper presents a model for managing the configuration and performance of the VC and VP services in these networks. The model consists of a connection management architecture that provides the services and is instrumented for network management purposes, and a network management architecture that uses the OSI management model to provide access to the appropriate service monitoring and control functions. The network management architecture proposes complete managed object definitions that achieve a variety of monitoring and control objectives: service configuration, service performance at the call and cell level, and service control. The latter includes the capability to establish VPs and control their allocated networking capacity and also control parameters of the signaling system that affect the performance of the VC service. The architecture was implemented in its entirety on the Xunet ATM testbed, and several experiments were conducted to illustrate the trade-off between the throughput of the VC service and the allocation of capacity to VPs. The experiments further demonstrated that the signaling system can create throughput bottlenecks in the absence of VPs in the network, and therefore, the network operator can increase performance by controlling the VP distribution.

Keywords: ATM, Service Management, Virtual Circuit, Virtual Path, Xunet.

1. Introduction

The significant progress achieved in the past few years in the development of signaling standards for ATM networks has generated a strong interest for architectures that enable the introduction and management of broadband services. This paper presents a model for managing the Virtual Circuit (VC) and Virtual Path (VP) connection services in ATM networks with Quality of Service Guarantees. These services can be regarded as the *core broadband services*, in the sense that other more complex services such as video-on-demand, video conferencing, virtual networks, etc., can be built on top of them. We give special attention to the VP service for the following reasons: VPs can be provided as an end-to-end connection service to the users or can be employed by the network operator to opti-

mize the call level performance of the ATM network. In this case, they can be used to establish connections for the VC service at a reduced signaling processing cost. This implies that the VP configuration (especially with respect to resource allocation) has a direct impact on the VC service performance. Therefore, the observed performance of the VC service should influence the decisions regarding the distribution of networking capacity to VPs (the *VP distribution*), and consequently, the VC and VP management architectures should be integrated.

In addition to the architecture for managing the VC and VP services, there is a need to develop an understanding of the architecture that provides these services, i.e., the connection management architecture. The VC service in ATM networks is provided through a signaling system, much in the same way that calls are established in telephone networks [BEL91]. VPs can be either installed from a management system or established through the signaling system. To this extent, we provide a connection management architecture for VC and VP connection setup under quality of service constraints. The architecture is integrated with the service management model. Integration here means that VCs and VPs set up by the connection management system can be instrumented for configuration and performance management purposes. The reader will quickly recognize that this instrumentation is representative for a large class of management problems such as billing (accounting), configuration management, etc.

1.1   Previous Work on Service Management

The review of service management efforts for ATM broadband networks reveals an increasing interest in the research community regarding the management of network services and especially of the VP service. [OHT93] presents an OSI-based management system for testing ATM Virtual Paths. The system is used exclusively for testing the cell-level performance of Virtual Paths, and allows the control of cell generators and the retrieval of information from monitoring sensors. The system is designed for testing purposes only and does not have the capability to install Virtual Paths, regulate their networking capacity, or measure any call-level statistics.

A more complete effort for standardizing the Management Information Base for ATM LANs that meets the ATM Forum specifications is currently under way in the Internet Engineering Task Force (IETF) [IET94]. This effort focuses on a complete MIB specification based on the SNMP standard for configuration management, including VP configuration management. Performance management is also considered but at the cell level only.

The ICM RACE project [ICM93] is defining measures of performance for ATM networks both at the call and at the cell level and the requirements for Virtual Path connection management. It is expected to deliver a set of definitions of managed objects for VP management and demonstrate an implementation of the architecture.

A more complete approach is taken by Griffin and Georgatsos [GRI95] who propose a system that offers the generic functions of performance monitoring, load monitoring and VP configuration management in ATM networks. In addition it provides functions for routing and bandwidth management in a hierarchical structure.

1.2   The Objective

The first objective of this paper is to describe the connection management architecture for establishing VC and VP connection services under QOS constraints. In essence, we provide a model for an

ATM signaling system that provides these services. The signaling system is then instrumented for the following service and performance management capabilities:

- Monitoring of individual Virtual Circuits for configuration information and cell-level performance.

- Control of individual Virtual Circuits such as renegotiation of the allocated resources or forcibly removing them from the network if, for example, they are a cause of congestion.

- Monitoring of the call-level performance of the VC service, such as call arrival rate, blocking probability, call setup time, etc., at several levels of abstraction. In contrast with the cell-level performance which is provided for every individual VC, the call level performance contains statistical averages computed by monitoring the total number of VC service requests.

- Monitoring of individual Virtual Paths for configuration information and cell-level performance. The later is computed from the cell-level performance measures of all the VCs that use the VP.

- Installation of new VPs in the ATM network or removal of existing ones

- VP configuration control, especially with respect to resource allocation.

- Control of operating parameters of the signaling system such as retransmission time-outs, call setup time-outs, etc., that affect the call level performance of the VP service.

- Monitoring of network links for configuration and resource utilization information.

- Control of the admission control policy at links and VPs.

These capabilities are encoded in Managed Objects which are then distributed in the MIBs installed on every agent in the network. In order to do this, an understanding must be developed of how these monitoring and control objectives are translated into attribute and behavioral information. This design procedure is empirical and cannot be formalized, and as a result, there exist many possible management system structures that accomplish the above monitoring and control objectives. We provide an informal methodology for deriving such a structure and verify our design on the Xunet III ATM broadband network that covers the continental United States.

The management system that we develop is then used to highlight how the VP service control capabilities can influence the performance of the VC service by obtaining measurements that reveal the fundamental trade-off between the throughput and the signaling processing load.

Since the scope of VC and VP management is very broad, we have identified the above management capabilities as a basic requirement for these services. Purpose of this work is not to claim that these capabilities will be sufficient for solving every management problem regarding these services, but rather to demonstrate how an abstract representation of management requirements can be mapped into a concrete service management architecture. To this extent, we believe that the work presented in this paper can serve as an example for solving a large class of service management problems.

This paper is organized as follows. Section 2 presents the architectural framework for installing and managing VC and VP connection services. Section 3 describes the Xunet III ATM testbed and other implementation details of the service management architecture. Network experiments with the objective of evaluating the management model and the performance of the network under several VP configurations are presented in Section 4. Finally, Section 5 summarizes our findings.

2. Architecture

This section presents an overall architectural framework for configuring and managing the performance of VC and VP services on ATM networks. Section 2.1 presents the Integrated Reference Model (IRM). The connection management architecture within the IRM is discussed in Section 2.2. The management architecture is outlined in Section 2.3. Finally, in Section 2.4, the integration of the service and network management architectures through the network telebase is presented.

2.1 The Integrated Reference Model

To overcome the complexity problems in emerging broadband networks - caused by the variety of communication services to be provided, the required quality of service guarantees, the large number of network nodes, etc. - there is an urgent need for integrating network management, connection management and real-time control tasks into a consistent framework. To this end, an overall model for network architectures was developed, called the *Integrated Reference Model (IRM)* [LAZ92]. In this model, the key role for network integration is played by the network telebase, a distributed data repository that is shared among network mechanisms.

The IRM incorporates monitoring and real-time control, management, communication, and abstraction primitives that are organized into five planes: the network management or N-plane, the resource control or M-plane, the data abstraction and management or D-plane, the connection management or C-plane and the user information transport or U-plane (Figure 1). The subdivision of the IRM into the N-, M- and C-plane on one hand, and the U-plane on the other, is based on the principle of separation between controls and communications. The separation between the N- and, M- and C-planes is primarily due to the different time-scales on which these planes operate.

The N- plane covers the functional areas of network management, namely, configuration, performance, fault, accounting and security management. Manager and agents, its basic functional components, interact with each other according to the client-server paradigm. The M-plane comprises the entities and mechanisms responsible for resource control, such as cell scheduling, call admission, and call routing; the C-plane those for connection management and control. The user transport or U-plane models the protocols and entities for the transport of user information. Finally, the data abstraction and management or D-plane (the Telebase) implements the principles of data sharing for network monitoring, control and communication primitives, the functional building blocks of the N-, M-, and C- and U-plane mechanisms. (A mechanism is a functional atomic unit that performs a specific task, such as setting up a virtual circuit in the network [LAZ93]).

2.2 Connection Management Architecture

According to our model, the VC service is provided to the users through a signaling system. Users provide their VC requests to the network through a network node referred to as the network *Point-of-Presence* (POP). Each request for the VC service is parametrized by traffic class, i.e., users can select any of the traffic classes available from the network (e.g., video, voice, etc.). Every traffic class provides different QOS guarantees, and so, the users must select the one that best matches their requirements. The POP then examines the parameters of the request and initiates a VC setup procedure through the ATM network. How the connection is routed is of little importance in this context. If there are available resources in the network, the connection is established and terminated at the des-
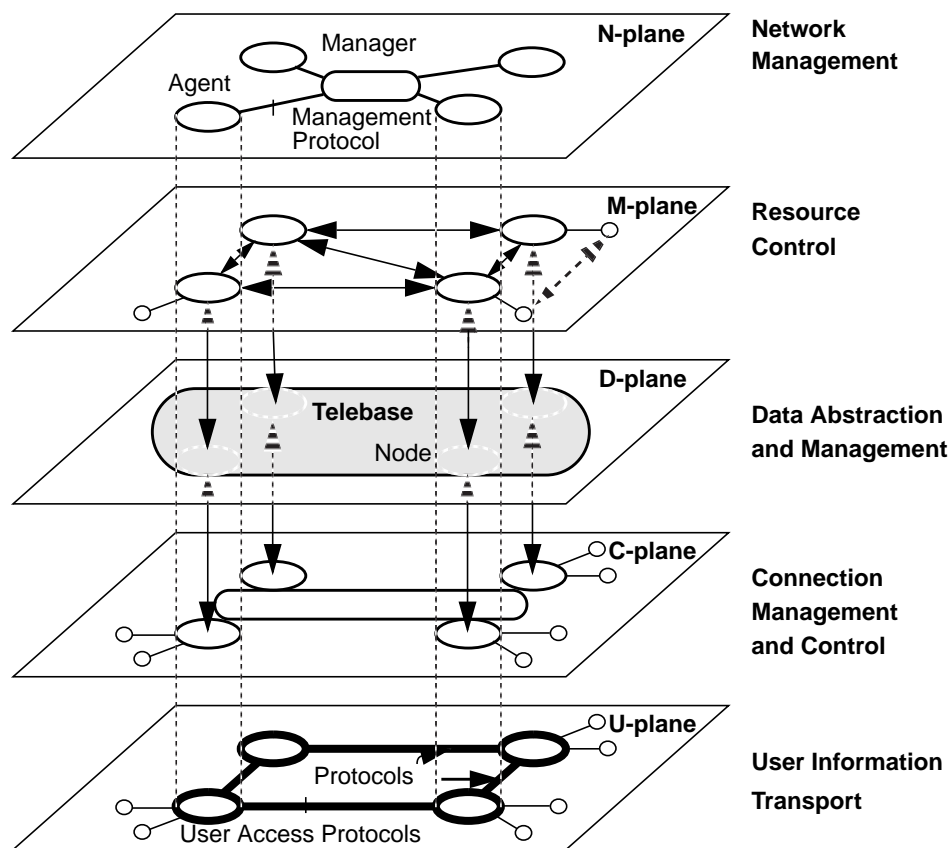
Figure 1: The Integrated Reference Model

tination user through a remote POP. Otherwise, it is blocked. If the request was successful, the originator (user) is notified that the connection has been established. Through the duration of the connection the user can renegotiate the allocated resources to the connection. The connection is terminated upon a user request or by force from the network operator (the network manager).

The VP connection management architecture can be divided in two parts: the first part describes a model for establishing VPs, and the second presents a model for VP operation during the VC setup procedure. In either case, central to the architecture is the D-plane of the IRM. The D-plane contains all information regarding the configuration and operational state of VCs and VPs and is used by the algorithms of the other planes both for monitoring and control operations.

The establishment of VPs is performed by the signaling system using a hop-by-hop setup procedure. The VP setup request can be presented by a user at a network POP, and in this case the resulting VP is terminated at the user (user-to-user or user-to-network mode). It can also originate from the network management system. This happens when a user does not have VP signaling capabilities with which to present a VP setup request to the POP, so he notifies the network manager to perform this operation on his behalf. The network manager can also establish VPs in network-to-network mode. In this case, VPs are terminated within the network and can be used for routing VC connections at reduced signaling costs.
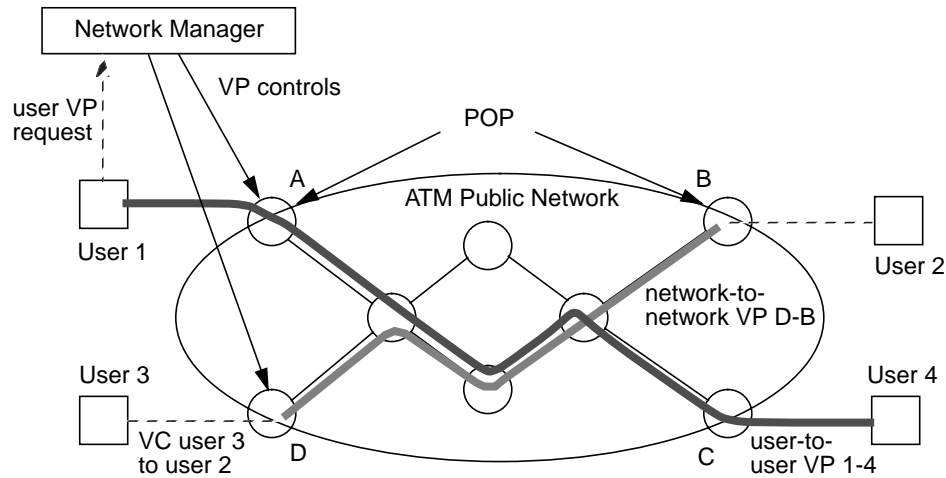
Figure 2: Virtual Circuits and Virtual Paths

Figure 2 demonstrates an example of the VC and VP setup procedure. The network manager establishes VP D-B in network-to-network mode by initiating the appropriate VP setup procedure at node (POP) D. This VP can be used to route VC connections requested at node D and destined for node B. For example, User 3 establishes a connection with User 2 by making a VC request at POP D. The ATM network then routes the requested VC connection over VP D-B and terminates the connection at User 2. Users can also establish VPs to other users by requesting the service directly from the POP. Alternatively, they can issue a request to the network manager to do this on their behalf. For example, in the above figure, User 1 wishes to establish a VP to User 4. He issues the VP request to the network manager and the manager using his VP control interface initiates a VP setup at POP A, indicating that the VP should be established in user-to-user mode (i.e., terminated at the customer gateway switch).

A VP is established along many physical network links. At every node along the route of the VP, the necessary networking capacity must be secured from the output link that the VP is traversing. In our model, the networking capacity of links is described by the Schedulable Region (SR) [HYM91], and of VPs by the Contract Region (CR) [HYM93b]. Informally, the Schedulable Region is a surface in a k dimensional space (where k is the number of traffic classes), that describes the allowable combinations of calls from each traffic class that can be accepted on the link and be guaranteed Quality of Service. The Contract Region is a region of the SR reserved for exclusive use by the VP. If the requested capacity allocation of a VP cannot be achieved, the allocated capacity at the end of the VP establishment phase is the minimum capacity available on the corresponding links (best effort resource allocation). If the allocated capacity is not satisfactory, the requestor has the option of releasing the VP. The set of all VPs in the network, characterized by their route, Contract Region and associated configuration information, comprise the *VP distribution policy*. The VP distribution policy is stored in the D-plane. Figure 3 shows how two VPs can be multiplexed on one physical link. We assume the existence of two traffic classes. QOS can be guaranteed if the sum of the two contract regions (which is a convolution operation) is a region which is strictly below the schedulable region of the link.
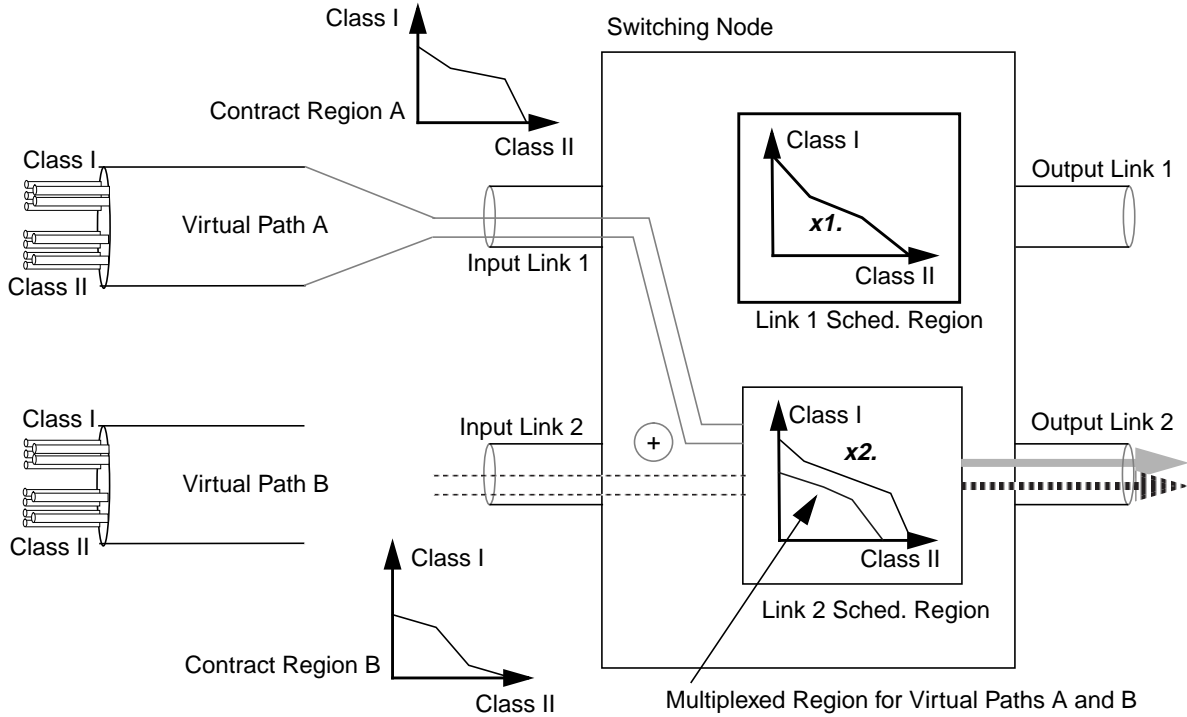
Figure 3: Multiplexing of VPs on the same physical link

An admission control algorithm located in the M-plane formulates the admission control policy (ACP), which is encoded as an object in the D-plane. The ACP is used by the signaling algorithm of the C-plane to make admission control decisions for incoming call requests.

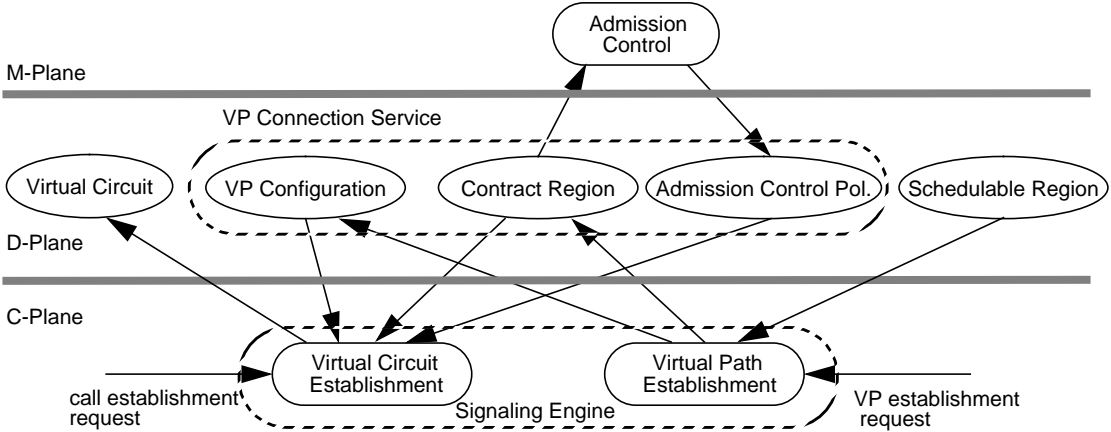Figure 4 shows the interaction between entities in the various planes of the IRM that provide the VP

Figure 4: Flow of Information during Installation and Operation of the VP Connection Service

connection service. During the VP establishment phase, the signaling engine creates a set of 3 objects in the D-plane: the CR, ACP and VP Configuration objects. The VP configuration object con-

tains general VP configuration information such as the input and output port numbers, the allocation of the VCI space, the VP operational state, etc.

During the VC establishment phase, the signaling engine reads the VP configuration object to determine if the VP can be used to reach the desired destination. It also reads the CR and ACP objects to examine if the call can be admitted on the VP. When the call has been established, a Virtual Circuit object is created in the D-plane that contains all necessary information for the VC. This information includes the VP Identifier (VPI) and VC Identifier (VCI), the traffic descriptor used to allocate resources, and other parameters for performance monitoring.

As mentioned in a previous section, VPs can be used in several ways. If the VP is terminated at the Customer Premises Equipment (CPE) (user-to-user or user-to-network mode), the customer is controlling the VP admission controller. In this case the VP can be regarded as a dedicated virtual link (or pipe) of a rated networking capacity. A network composed of such VPs terminated at the customer premises is also known as a Virtual Network (VN). The Network Manager has the capability to configure and maintain a VN by managing the individual VP components according to the customer requirements.

Alternatively, the termination of VPs may not be visible to the network customer. In this case (network-to-network mode), VPs are used by the network operator to improve the performance of the signaling system, the availability of resources between a pair of nodes, or even improve certain call level measures of Quality of Service for the users, such as call setup time and blocking probability. The existence of network-to-network VPs is completely transparent to the users.

2.3   Network Management Architecture

The Management Architecture builds on the OSI Management model [ISO88]. According to this model, network entities (either physical, like hardware modules, or logical, like virtual circuits) are mapped into "Managed Objects" for monitoring and control purposes. The managed objects are also referred to as *logical objects* and the network entities that they represent as *real objects*. A Management Agent contains the information about the managed objects in the Management Information Base (MIB). Managed objects are characterized by a set of attributes that reflect the state of the corresponding real object and behavioral information, which defines the result of management operations on the managed object. A proprietary protocol can be used for *linking* the state of every real object to its logical counterpart in the MIB. The Manager connects to the agent(s) and performs operations on the objects in the MIB using CMIP (the Common Management Information Protocol). These operations are of synchronous nature, i.e., they are initiated by the manager who, then, waits for a reply from the agent(s). Events of asynchronous nature (notifications) such as hardware faults can be emitted from the agent(s) to the manager using the event reporting primitive of CMIP.

Management operations take place in the N-plane of the IRM (Figure 1). The MIB of every agent is located in the D-plane of the IRM. As a result, the linking of the logical objects in the MIB with real objects is done within the D-plane [TSU92]. Control operations from the manager applied to objects in the MIB are reflected in the state of the real objects of the D-plane, which in turn, affect the behavior of the algorithms in the C- and M- planes. Conversely, the change of state of the real objects in the D-plane will cause an update of the state of the logical objects in the MIB.

The question that arises is how to define the appropriate managed objects and their attribute structure that will satisfy the monitoring and control requirements of the management problem at hand. As mentioned previously, this procedure cannot be formalized. However, the manager can follow the following rules of thumb:

- In object-oriented modeling, a system can be represented by defining logical equivalents (objects) of the physical entities that will be managed. For example, the MIB in our service management system can contain objects that relate to Virtual Circuits, Virtual Paths, physical Links, Hardware Modules, etc.

- The quantities that need to be monitored and controlled in each one of these objects can be represented as one of their attributes. For example, the cell loss probability for a VC can be represented as one of the attributes of the corresponding managed object.

- Every attribute is assigned a "behavior". The behavior indicates the semantics of management operations (such as `get` and `set`) on the attribute. For example, every VP object contains an attribute representing the contract region (CR) of the VP. Using the behavioral description, the agent knows how to obtain the value of the contract region from the D-plane upon a `get` operation. When the CR is modified with a `set` operation, the agent knows also how to propagate this change in the D-plane, and then to the signaling system, which in turn will initiate a procedure to modify the allocated capacity of the VP.

- There may exist objects in the managed system that do not represent physical network entities. For example, our model needs to monitor the call-level performance of the VC service between a source and a destination POP. In this case, a managed object can be defined with attributes whose value is obtained from the appropriate state and event information in the D-plane. In this example, performance measures such as call arrival rate, etc., can be extracted by filtering event information in the D-plane related to VC setup requests. The filter is such that only VC requests between the POP pair are taken into account.

Managed object attributes are therefore either a reflection of an attribute of an object in the D-plane, or the result of processing and filtering operations on several D-plane objects and event information. Therefore, for every managed object attribute, the management system implementor must provide information on how the value of the attribute is obtained from the D-plane, or, in the case of control operations, how the new attribute value is propagated in the D-plane. This procedure is also known as the *state linking* of the physical and the logical objects, and must be coded in every agent of the management system.

Therefore, in our model, monitoring and control of the VC and VP services is possible by defining the appropriate managed objects in the MIB and linking them to the appropriate entities of the D-plane. What managed objects to define, how to integrate them in the D-plane and how to define their behavior will be the topic of the following section.

2.4   Integration of the Connection and Network Management Architecture

The purpose of this section is to describe the object level model for VC and VP management and its integration within the D-plane. Management operations take place in the N-plane. Our objective is to translate the management system requirements that were outlined in Section 1.2 into a management cycle and then into concrete managed object definitions.

2.4.1 The Management Cycle for VC and VP Management

One of the objectives of service management is to ensure that services are provided to the users with Quality of Service guarantees. In order to evaluate the offered service QOS, the manager must first define a set of service performance measures. [NEU93] defines two different levels of refinement for performance management purposes: the call level and the cell level (Figure 5). On the call level, we
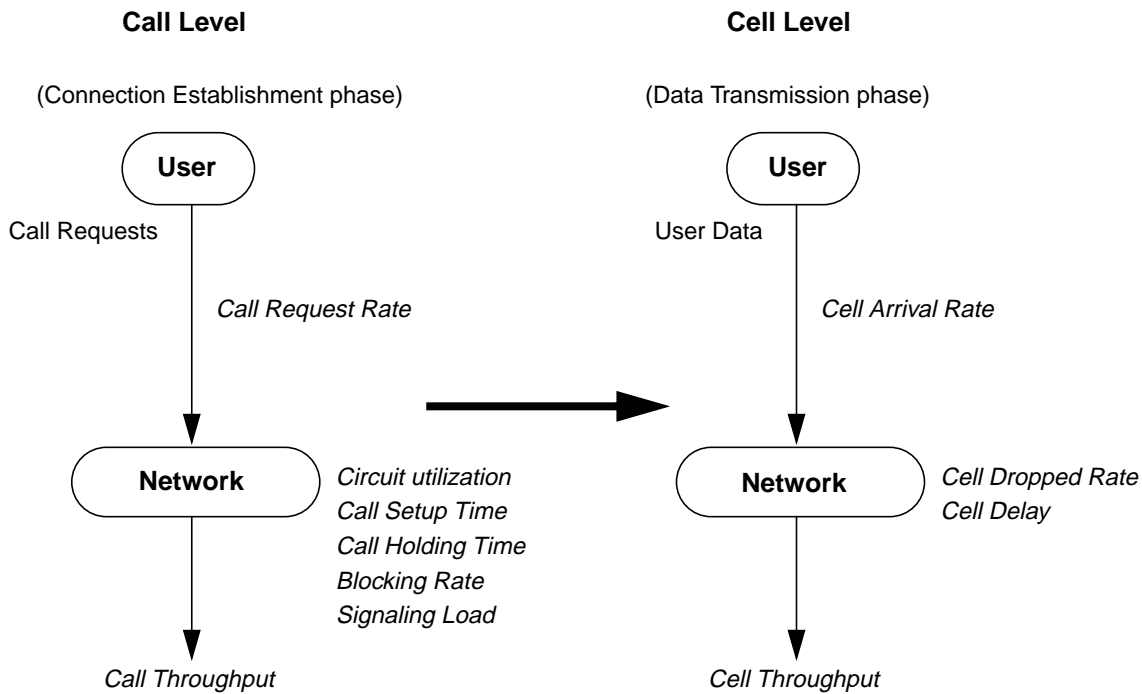
Call Level   **Cell Level**

(Connection Establishment phase)   (Data Transmission phase)

Figure 5: Performance Management Related Properties at the Call and the Cell Level

consider the ATM network as the service provider. The user makes service requests by attempting to establish VC connections of a certain traffic class (e.g., voice, video, etc.). The call level performance measures include the following:

- Call request (arrival) rate: average number of connection requests in the unit of time
- Utilization and capacity: number of currently open connections and maximum number of open connections allowed (due to limits in the available networking capacity or availability of VC identifiers).
- Service time: average holding time of connections. The service time multiplied by the request rate gives the offered load in Erlangs.
- Blocking (or Error) rate: the rate of calls being blocked due to resource unavailability or other errors. Further subdivision of this quantity into categories can give more information on the rates with which every error occurs. For example we can measure the rate with which connections are being blocked due to networking capacity unavailability, routing errors, virtual circuit identifier unavailability, signaling errors, etc. The ratio of the error rate with the request rate gives the blocking probability.

■ Signaling load: this quantity measures the load imposed on the signaling system and depends on the call request rate and the necessary time to complete the connection setup request. The higher the load, the less the ability of the signaling system to provide the service. Factors that contribute in increasing the signaling load are: high number of signaling message retransmissions due to protocol errors, higher percentage of connection attempts over routes with many hops (that increases the number of transmitted messages), high call arrival rates, etc.

The second level of refinement for performance monitoring purposes refers to the data transmission phase of communication, after the connection has been established. The user and the network maintain the service user / service provider relationship but with different semantics: the user sends cells to the network for transmission. The following performance related properties are of significance:

■ Cell arrival rate: average number of cells arriving from the user in the unit of time.

■ Cell drop rate: average number of cells that are being dropped in the unit of time.

■ Cell throughput: average number of cells transmitted successfully.

■ Cell transmission delay: average delay for to transmitting of a cell from the time it has been received at the input port.

We can see that cell level measurements are related to the performance of individual VC connections. The call level measurements however are not tied to a specific connection but represent statistical performance measures for the VC service. Further, the call level measurements can be computed at different levels of abstraction by considering only the appropriate subset of VC service requests. For example, we can measure the call level performance by isolating the VC requests between a source and a destination POP (also referred to as a Source-Destination pair - SDP), or by considering only the requests offered to a specific link admission controller, etc.

The manager observes the above service performance measures and determines whether services meet their QOS requirements. The objective at the call level is to provide a bounded call blocking probability and call setup time for every SDP in the network. Blocking can be caused by several factors including a faulty link, a poor VP capacity allocation policy, a routing malfunction, an overloaded signaling processor, etc. High call set up times may prompt the user to cancel his request, leading to loss of revenue for the service provider. At the cell level, the objective is to guarantee the negotiated QOS parameters. We assume that cell level QOS is provided by the network traffic control functions that operate in real time, and therefore does not require the intervention of the service management system. We also assume in the sequel that users make requests for the VC service only, i.e, VPs are employed only by the network manager in network-to-network mode.

The manager can satisfy the above call-level QOS objectives by controlling the VP distribution in the network and other parameters of the signaling system. In more detail, the manager has available the following set of controls:

1. Create a VP.

2. Delete a VP.

3. Change the Contract Region of a VP.

4. Change the Admission Control Policy of links or VPs.

5. Control other parameters of the signaling system, such as the time-out for call setups, the time-out for message retransmissions and the signaling congestion thresholds (the latter enable the congestion control mechanisms in the signaling system during periods of overload).

Summarizing, the management cycle for the VC and VP services consists of monitoring the call level performance of the network at different levels of abstraction and producing a VP distribution to fit the offered load to the network while satisfying the call-level QOS constraints. Informally, increasing the capacity allocated to a VP results in reducing the signaling load on the network, but also in a possibly reduced network throughput. In addition, the manager may control some signaling system parameters that affect the call level performance. The cycle then starts from the beginning.

2.4.2 The VC and VP Management Object Model

This section provides the managed object definitions that satisfy the monitoring and control requirements outlined in the previous section. In object-oriented modeling, a *class* represents a set of objects with the same characteristics. After defining the generic classes of network objects, every concrete object is represented as an instance of one or more of them. At the definition phase, we can define various managed object classes efficiently using the attribute inheritance mechanism, in a step by step specialization from a general and abstract level of managed object classes to more specific ones. As a result, every network managed object class becomes a node of a managed object inheritance tree. This tree represents all inheritance relationships among managed object classes. Our modeling follows the OSI guidelines for the definition of managed objects (GDMO) [ISO91b]. Table 1 provides a mapping of the requirements of Section 1.2 into managed objects.

Table 1: Mapping of the Service and Performance Management requirements into Managed Objects

Service Management requirement	Managed Object
Link Configuration	Link object
Link Admission Control	Link object contains an admissionControllerPackage
VC configuration monitoring	VCEntity object contains attributes with configuration information
VC cell level performance	VCEntity object contains a cellPerformancePackage with the cell level performance attributes
VC service performance	Link, Source-Destination and VP objects contain a callPerformancePackage with the call-level performance attributes
VP configuration monitoring	VP object
VP Create	Agent creates a VP object with the supplied configuration information
VP Delete	Agent deletes a VP object
VP capacity change	Contract Region is a settable attribute of the VP object
VP Admission Control	VP object contains an admissionControllerPackage
Signaling System Control	ConnectionMgmt object

According to our model, one agent is installed on every ATM switch. The agent processes the information on call attempts from every input port. For each successful call attempt, an object of class

VcEntity is created for the corresponding Virtual Circuit connection (VC). Each VC object contains configuration information such as the number of the input and output slot, Virtual Path Identifier (VPI) and Virtual Circuit Identifier (VCI). In ATM terminology, this implies that the VC object models the input and output Virtual Circuit Link (VCL) at every switch. Thus, the end-to-end management of a VC that spans many switches (and hence has one instance in each OSI agent at every switch) is achieved by managing the individual objects in combination. Additional attributes for each VC include the call establishment time, traffic descriptor (composed of a service class characterization and the allocated networking capacity in kilobits per second), adaptation layer information and source/destination end-user service information. The package *cellPerformancePackage* contains attributes associated with the cell-level performance related parameters, and will be described below.

The class *VirtualPath* derived from *Top* is used to describe a VP. The VP object in analogy with the VC object is comprised of an incoming and an outgoing part at every switch. At the VP source or termination point, the VP has only an outgoing / incoming part respectively. Attributes used to describe the configuration of the Virtual Path are: *vpIdentifier* (VPI), *vpSource*, *vpDestination* (VP source and termination address), *circuitCapacity* and *timeEstablished*. The VP object at the source also contains a *callPerformancePackage*, and an *admissionControllerPackage.* These will be described below.

The class *Link* is derived from *Top* and is used to model input or output network links. The mandatory attributes for this class are *linkType* (input or output), *linkModuleDescription* (describes the hardware of the link interface), *linkSource, linkDestination* and *linkSlotNumber* (the slot number the link is attached to). If it corresponds to an output link, it contains a *callPerformancePack*age, and an *admissionControllerPackage*.

The class *SourceDestination* is used to describe the call level activity between a pair of nodes, and can be used to evaluate the call level performance in an end-to-end fashion. A Source-Destination (SD) object exists in the agent if there is call-level activity between the two nodes, and the source node is either the local switch, or a directly attached User-Network Interface (UNI). The SD object contains the following attributes: *sourceNodeAddress* and *destinationNodeAddress* and a *callPerformancePackage.*

The *callPerformancePackage* is an optional package that measures the call-level performance. It is contained in all SD objects, and in some link and VP objects. For the objects of class *Link,* the package measures the activity for calls that follow the link but not a VP that uses the same link. For VP objects, the package measures the activity of call requests that use the VP. The attributes of the *callPerformancePackage* are the following: *activeCircuits*, *callArrivalRate* (average arrival rate of requests in calls/min), *callArrivedCounter* (counter of call requests), *callResourceBlockedCounter* (counter of calls blocked due to resource unavailability), *callErrorBlockedCounter* (counter of calls blocked due to protocol errors, e.g., time-outs, etc.), *callBlockingRate* (average rate of calls blocked for any reason in calls/min), *setupTime* (average time to establish the connection in milliseconds), *holdingTime* (average duration of connections in seconds), *numExchangedMessages* (average number of messages that have been exchanged to setup the connections, as an indicator of the processing required for each connection), and *measureInterval* (the time in which the above averages are computed in seconds). All quantities are measured separately for each traffic class, and then a total over all classes is computed.

The *cellPerformancePackage* measures cell-level performance. The attributes *cellTransmitted-Counter, cellTransmissionRat*e, *cellDroppedCounter* and *cellDroppedRate* measure the number of cells transmitted or blocked and their respective time averages. The attribute *avgCellDelay* measures the average time from the reception till the transmission of cells from the switch. The package is included in objects of class *VcEntity*, and in this case, only the cells belonging to the VC are measured. As an option, it can also be included in objects of class Link, SourceDestination or VirtualPath. In the latter case, a sum of the attributes over all VC objects that belong to the Link/SourceDestination/ VirtualPath is computed, and the respective attributes of the Link/SourceDestination/VirtualPath objects are updated.

The package *admissionControllerPackage* is mandatory for output link and VP objects. It describes the state of the admission controller, which is located at the output links (for switches with output buffering) and at all VP source points. The package contains the following attributes: *networking-Capacity* (the schedulable region for link objects or the contract region for VP objects), *admission-ControllerOperatingPoint* (the operating point of the admission controller given the established calls for each traffic class), *admissionControlPolicy, admissionControllerOperationalState* (enabled (call requests allowed to go through and allocate bandwidth) or disabled) and *admissionControllerAd-ministrativeState*.

The class *ConnectionMgmt* contains attributes that control the operation of the local signaling entity. There is only one instance of this class in every agent. The attributes of this class are the following: *signalingProcessingLoad* (an index of the call processing load observed by the signaling processor), *maxSignalingProcessingLoad* (the maximum signaling load value allowed, beyond which the signaling processor denies all call establishment requests), *signalingRetransmitTimeout* (the time-out value in milliseconds for retransmitting a message if no reply has been received), and *signaling-CallSetupTimeout* (the maximum acceptable setup time in milliseconds for a call establishment. If the time to establish a circuit is more than the current value, the circuit is forced to a tear-down). The single instance of this class is also used to contain four other container objects of class *LinkMgmt*, *SourceDestinationMgmt*, *VirtualPathMgmt*, and *VirtualCircuitMgmt*. There is only one instance from each of these four classes, which is used to contain all objects of class *Link*, *SourceDestination*, *VirtualPath*, and *VirtualCircuit*, respectively.

Figure 6 shows the inheritance tree of the object model. For simplicity, we omitted the definition of
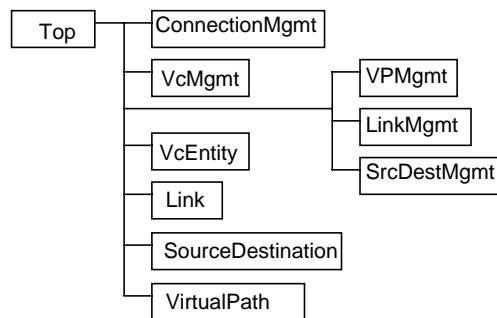

Figure 6: The Inheritance Tree

associated thresholds for each performance variable that can trigger notifications in case of threshold crossing [ISO92]. Such definitions can be easily incorporated in the above model.

As discussed in the previous section, the MIB of every agent resides in the D-plane. Managed Objects use the information stored in the D-plane to update their state. Figure 7 demonstrates this prin-
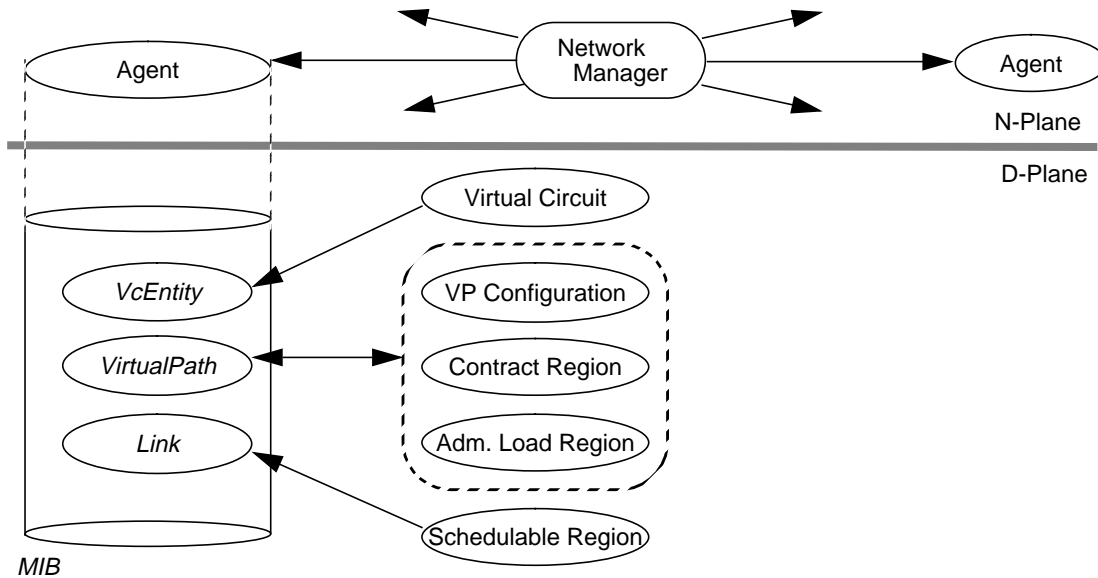

Figure 7: The Integration of the Connection and Network Management architectures in the D-plane

ciple. For example, the Managed Objects of class *VcEntity* represent the Virtual Circuit object that was created in the D-plane by the signaling system. The attributes of the managed object mirror the state of the corresponding real object. In the same manner, the MO of class *VirtualPath* contains attributes that reflect the state of the corresponding real objects (VP Configuration, Contract Region and Admissible Load Region). An MO of class *Link*, uses the object Schedulable Region (among other information), to reflect the state of the link Schedulable Region on one of its attributes. Additional processing of events (such as VC creation, etc.) inside the agent can provide the necessary call-level performance related properties (such as call arrival rates). These might not be readily available from other objects of the D-plane.

3. Implementation of the VC and VP Management Architecture

3.1 The Xunet III ATM Testbed

Xunet is one of the five Gigabit testbeds sponsored by the Corporation for National Research Initiatives. It has been deployed by AT&T in collaboration with several universities and research laboratories in the continental United States [FRA92]. The topology of the network is shown in Figure 8. The network links are rated at 45 Mbps and 622 Mbps. Access at every node is provided by 200 Mbps network interfaces. A workstation serves as the switch Control Computer (CC) at each network node. The CC runs the switch control software that performs signaling, control and fault detection functions.
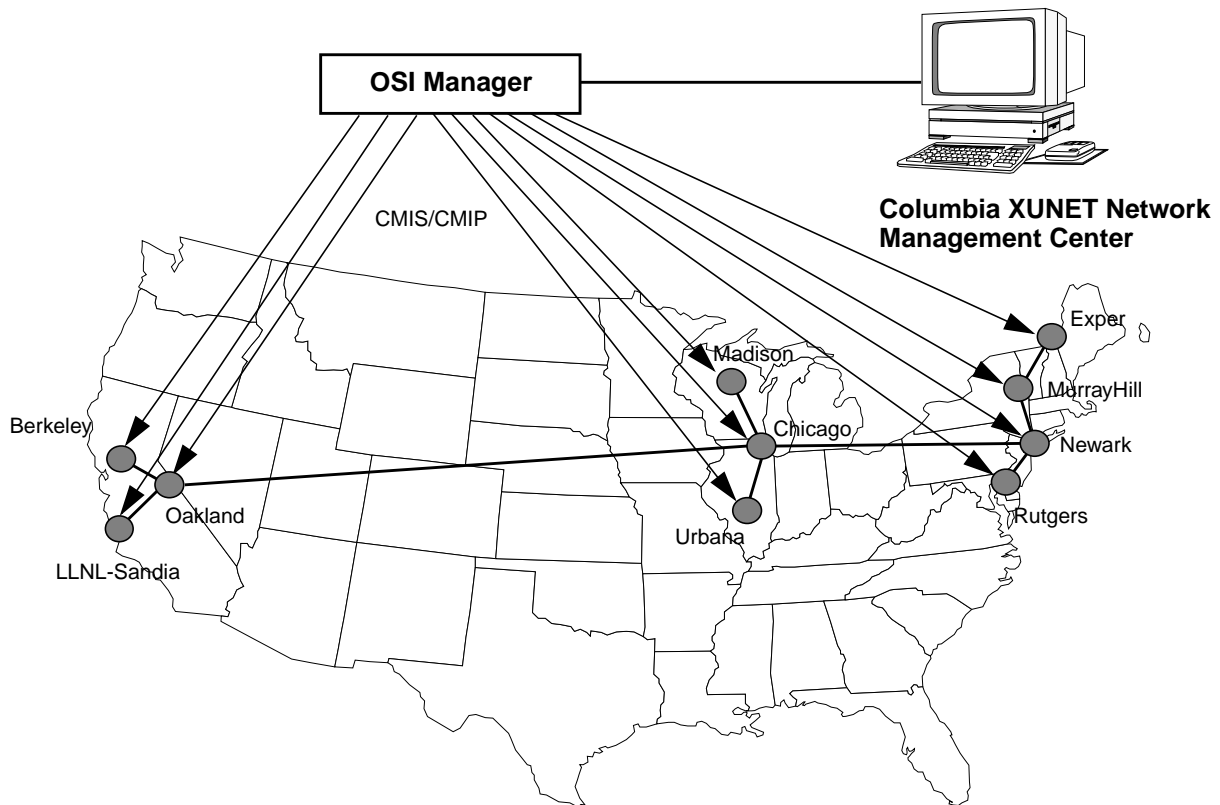
Figure 8: The Xunet Management System

3.2   The Xunet Signaling and Admission Control Architecture

Xunet supports five traffic classes. Class 1 is used for high priority control messages and is given absolute priority by the link scheduler (multiplexer). Class 2 is used for Video service, Class 3 for Voice, Class 4 for priority data and Class 5 for bulk data [SAR93].

A signaling system very similar in characteristics with the CCSS#7 (Common Channel Signaling System) has been installed on Xunet. The system allows virtual circuit establishment with best effort resource allocation in a single pass. An admission controller operates on every output link. The necessary Schedulable Region and Admission Control Policy objects are downloaded from a management station. The admission control policy used is Complete Sharing [HYM93a].

Virtual Path establishment is also done in one pass with best effort resource allocation. When the VP has been established, an admission controller is activated at the source node of the VP that uses the allocated contract region for admission control decisions. The admission control policy is also Complete Sharing. A signaling channel is also established between the two VP termination points to carry VC establishment requests over the VP. It operates in the same way as the signaling channel used on every physical link. As a result, from the point of view of the signaling system, VPs are considered as regular links with only minor differences.

Every Contract Region can be changed dynamically. The deallocation or allocation of additional resources is performed in the same way as in the VP establishment phase. Finally, when a VP is re-

moved, the Contract Region is returned to the Schedulable Regions of the links along the path, all VCs using the VP are forced to termination and the VP signaling channel is destroyed.

3.3 The Xunet OSI Management System

From the five functional areas covered by the OSI management model, we have chosen to support configuration, fault and performance management for Xunet (the remaining functional areas being security and accounting management). The fault management architecture enables us to monitor closely all the network switches for hardware faults, such as link level errors, buffer overflows, etc. The configuration management architecture allows us to monitor and control the hardware configuration of the switch and the configuration of the VC and VP services. The performance management architecture provides the service performance measures outlined in Section 2.4.

As the basis of our OSI Management system, we have selected the OSIMIS software [KNI91]. Our implementation expanded the agent with managed objects for Xunet and the management applications to include powerful graphics that depict the operational state of the network and a point-to-multipoint control interface that facilitates the overall management task [ANE93]. The management applications run at Columbia University. TCP/IP sockets are used at the transport layer to connect to the agent at each site. Inside the agents, communication between logical and physical objects is achieved by using a proprietary protocol between the agent and the Xunet switch. For this purpose, we use UDP/IP packets over a local Ethernet. The structure of the system is shown in Figure 9.
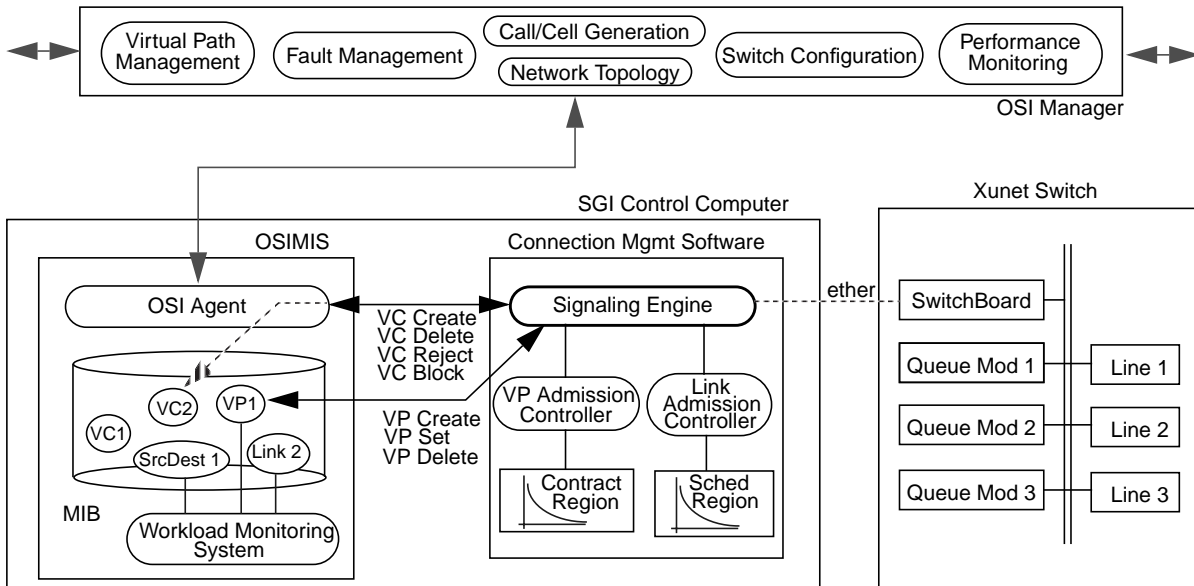

Figure 9: Structure of the Xunet Management System

3.4 The OSI Agent

The OSI agent contains managed objects for configuration, fault and performance management. Figure 10 shows the containment tree for the Xunet MIB. In contrast with the inheritance tree, the containment tree depicts the organization of managed objects in the MIB. Its main usage is in the definition of the Distinguished Names (DNs) that specify each managed object instance. However, since the containment relationship specifies only a naming relationship, there exist many ways of
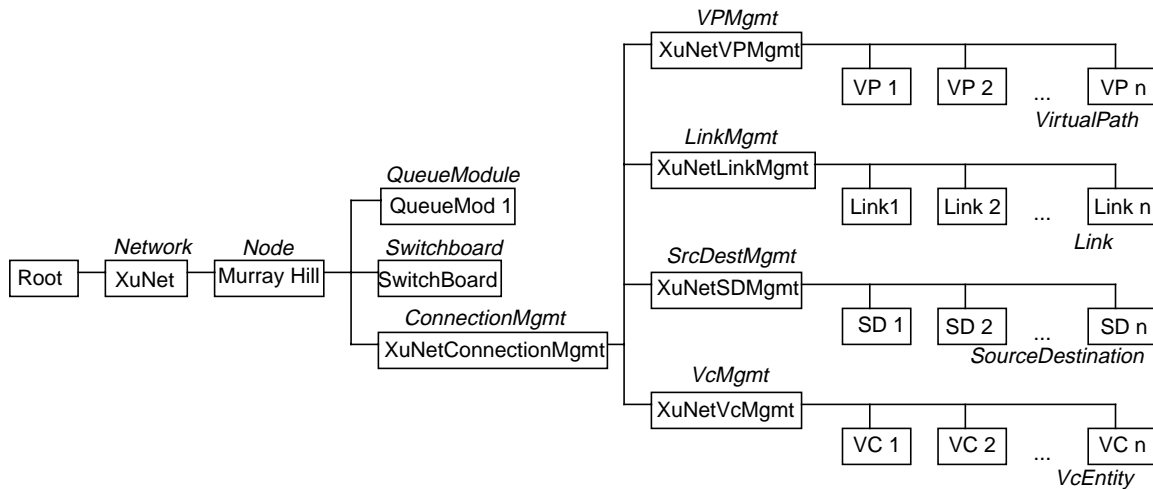
Figure 10: The Containment Tree of the Xunet MIB

defining a containment tree for the management system in hand. There exists no systematic way of determining if one containment tree is better than another, and as a result, the structure of the containment tree does not have immediate implications in the capabilities of the management system. The Xunet containment tree consists logically of two major groups of Managed Objects.

3.4.1 Hardware Configuration and Fault Management Group (HCFMG)

For the purpose of configuration and fault management, we have implemented managed object classes for each Xunet hardware module, such as *SwitchBoard*, *QueueModule*, etc. Each module is polled at regular time intervals by the agent to detect possible faults. A hardware fault triggers a notification inside the agent, which in turn can generate a CMIS Event Report primitive if the appropriate Event Forwarding Discriminator object has been created by the manager [ISO91a]. Currently, more than 300 different hardware errors can produce an equal number of event reports. This wealth of information provides the manager with extensive fault monitoring capabilities. The configuration and state of the hardware modules is obtained from the Xunet switch every 20 seconds. The information is processed internally to update the corresponding managed objects.

The set of the hardware managed objects also gives complete configuration information of every switch. The management applications can display graphically the configuration and indicate the location of every generated event report.

3.4.2 Service and Performance Management Group (SPMG)

The SPMG consists of a set of managed objects that monitor and control the service configuration and performance of Xunet both at the cell and at the call level. All call level information is obtained from the local signaling entity. The OSI agent receives four types of events: VC-Create, VC-Delete, VC-Blocking (no network resources) and VC-Rejection (any other cause), with the appropriate parameters. Upon a VC creation event, a Managed Object of class *VcEntity* is created inside the MIB that contains all the available information on this VC. The object is related to the appropriate *Link*, *SourceDestination* or *VirtualPath* objects. Every 30 seconds, the Xunet switch is scanned to compute

the number of cells transmitted or dropped for each VC. At the same time we use this information to update the total number of cells transmitted or lost on a link, SD pair or VP based on the relations defined when the VC was created. The VC object is removed by a deletion event.

All 4 event types cause the agent to update some internal counters in the corresponding Link/SD/VP objects. Additional processing is performed at a regular time interval (controllable by the manager through the *measureInterval* attribute, and usually set to 30 seconds). At that time, the agent calculates the values for the performance related attributes, taking into account only the events that occurred during the past interval. For example, when a VC is created, a counter that registers the call arrivals is incremented. At the end of the 30 second period, the arrival rate is calculated, and the counter is reset to zero. All other attributes are calculated in a similar fashion. Figure 11 shows the
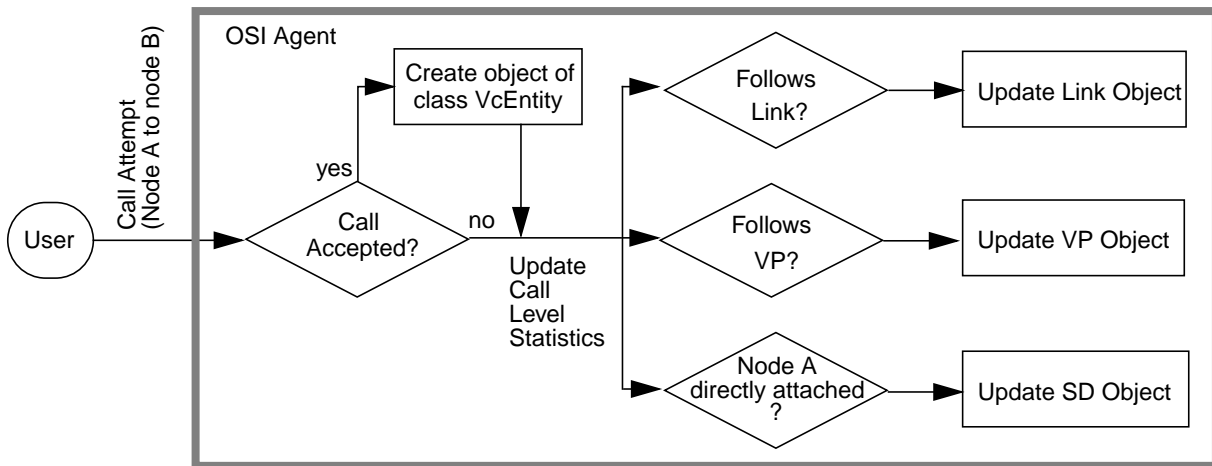

Figure 11: Flowchart for computing the call level performance

procedure for creating VC objects and updating the objects that contain the call-level performance package.

VP management functions originate at the network manager site. When the management application issues a `create` command with the appropriate parameters, a VP managed object inside the MIB is instantiated, and the Xunet signaling entity is informed to initiate a VP setup procedure. VPs can be subsequently modified by the `set` command operating on the appropriate object, and deleted with a `delete` command.

Parameters of the signaling entity are controlled through `set` operations on attributes of the *ConnectionMgmt* object. Each `set` operation causes a control message to be sent from the agent to the signaling entity.

3.5 The OSI Manager

Xunet is monitored and controlled through a Motif/X-Toolkit-based application. The same application is used to run controlled call and cell generation experiments on Xunet. It consists of six tightly interconnected subsystems (Figure 9). Every subsystem contains the appropriate display tools and management interface functions for a specific management task:

1. Switch Configuration: Displays the hardware configuration of the switch using information from the objects of the HCFMG.

2. Fault Management: Receives OSI Event reports from the agents, that are related to hardware problems, and uses the Switch Configuration subsystem's display functions to inform the manager about the nature and location of the problem.

3. Network Topology: Displays a map of the network, with all switches, links and attached user-network interfaces. The displayed objects can be selected and observed independently. Support is also provided for displaying the route and configuration information of VPs.

4. Virtual Path Management: The manager is able to create and subsequently control VPs with `create` and `set` operations. The VP control task is guided by the observations obtained from the Performance Monitoring system.

5. Performance Monitoring: Collects the information that is provided by the SPMG objects in each node and displays it using the functions of the Network Topology subsystem. The information can be either displayed in textual form, or graphically. In the latter case, we use time series plots that are updated in real-time. The plots allow us to observe the performance "history" of the network and the effects of VP management controls.

6. Call and Cell Generation: The Xunet signaling entities contain a call generation facility. A managed object inside the local agent makes it possible to control the call generation parameters in terms of destination nodes, call arrival rate and call holding time on a per traffic class basis. The call generation system can also be linked to the Xunet cell generator for real-time cell generation [ANE95].

Figure 12: The Xunet Management Console displaying the call level performance

In addition to supporting the elementary management operations on managed objects (such as `get`, `set`, `create`, `delete`, `action`, etc.), the management application supports the following primitives for network-wide management operations:

1. Computation of network-wide call-level statistics by collecting information from all SD objects scattered across the network.

2. Installation of the VP distribution policy in the entire network.

3. Activation and deactivation of all call generators in the network.

4. Performance

We have used the management system to run controlled experiments on Xunet to study the call level performance of the network, such as the performance of the signaling system and the network throughput under various VP distribution policies. Call level experiments consist of loading the signaling system with an artificial call load. A Call Generator on every switch produces call requests with exponentially distributed interarrival and holding times. The call-level experiments have been separated in two phases. In the first phase, we measure the performance of the signaling system without using VPs. This experiment allows us to find suitable values for the congestion control parameters of the signaling entities that give the highest call throughput. The second phase builds upon the first phase and attempts to determine the call throughput by measuring the performance of the network with VPs in place.

4.1   Performance of the Signaling System for Virtual Circuit Set-Up

In this experiment, the network was loaded with an artificial call pattern. Our goal was to measure the performance of the signaling system under heavy call arrivals. For each call, the signaling system sets up a VC by traversing all the nodes in the path and patching the appropriate connections in each node. Call generation is controllable for each destination in terms of the call arrival rate and call holding time for each of the five traffic classes supported by Xunet. The network was homogeneously loaded from five sites (Murray Hill, Rutgers U., U. of Illinois, U.C. Berkeley and Livermore) with Poisson call arrivals and an exponential holding time with a mean of 3 minutes. We used only video calls (assumed to consume a peak of 6 Mbps/call) and voice calls (64 Kbps/call) in a ratio of 1:10 (i.e., the arrival rate of voice calls is 10 times greater). All the links in the experiment described here are of 45 Mbps capacity. The schedulable region (SR) of each link assumed to be given by a two dimensional hyperplane. We used peak rate allocation, and according to this scheme, the SR can accommodate a maximum of 7 video calls or 703 voice calls. The admission control policy used was complete sharing [HYM93a].

Figure 13 shows the measurements obtained by gradually increasing the call generation rate. Each measurement (throughput, call setup time, call blocking, etc.) is computed by adding (or averaging) the respective measurements for the video and voice traffic. Both the call throughput and call blocking due to resource unavailability (the "Throughput" and "Blocked Percentage" curves) rise almost linearly with the call arrival rate. The sudden drop in the total call throughput is due to the overloading of the signaling system with call setup requests. At that point, the call setup time and the percentage of calls blocked due to congestion of the signaling system (the "Rejected Percentage" plot) start to rise sharply. The "Blocked Percentage" curve drops because the strain has now been moved from network transport to call setup, and thus, calls contend for signaling resources rather than networking resources. During overload, only a small percentage of the total call attempts is actually established, and therefore, the probability that these calls will find no networking capacity available is diminished. In the extreme situation, all calls are blocked while the networking capacity of all links is unused.
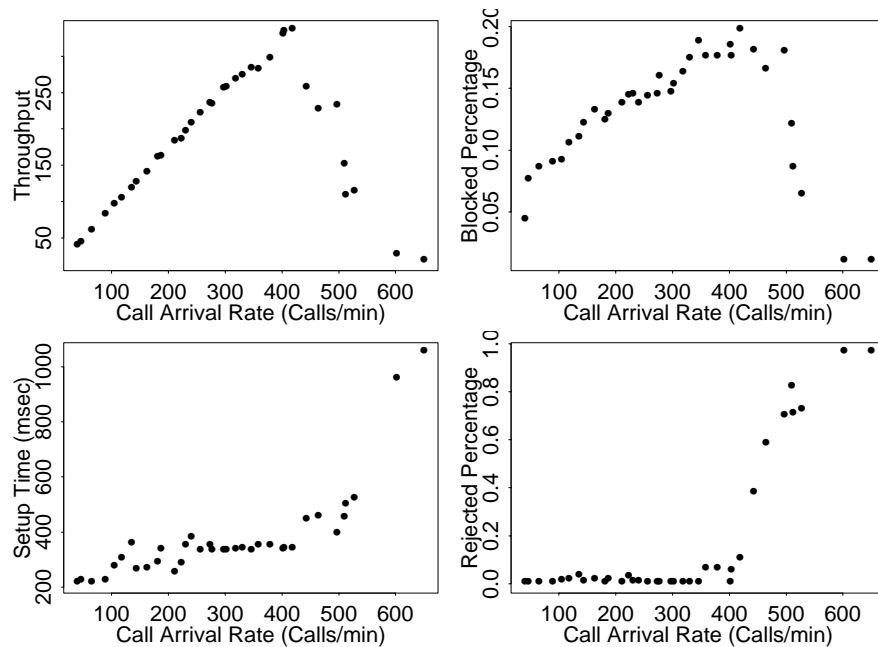
Figure 13: Performance of the Signaling System

The congestion situations seem to appear first at the Newark and Oakland switches, that are the first to become overloaded with call request messages. It is therefore essential for the network manager to regulate the call arrival rate at the entry points in the network. This can be done by setting an appropriate value for the *maxSignalingProcessingLoad* attribute of the ConnectionMgmt object. The signaling load is computed from the number of signaling messages received and transmitted from the switch in the unit of time. If the load reaches the *maxSignalingProcessingLoad* value, a fraction of the incoming call requests are discarded. We have found experimentally that by restricting the signaling load to about 450 messages per minute at the nodes connected to the call generators, the network operates within the capacity of the signaling processors.

4.2 Performance Trade-off due to VP Capacity Allocation

This experiment had the objective of studying the trade-off between the network throughput and the allocation of networking capacity to VPs. It was performed on the east coast segment of the network. This four node segment consists of two nodes in MurrayHill (called MHEX and MH), one node in Newark (NWRK), and one at Rutgers University (RUTG) connected in tandem.

The generation ratio between Class 2 and Class 3 calls was 1:100. The call arrival rate from each call generator was kept fixed throughout the experiment. The generator at MHEX produces traffic to NWRK (180 calls/min) and RUTG (210 calls/min). The generator at MH produces only traffic to RUTG at 180 calls/min. The generator at RUTG produces traffic to NWRK at 180 calls/min. One VP is established from MHEX to RUTG (Figure 14).

Only the traffic from MHEX to RUTG is allowed to follow the VP. Calls that find the VP full and the calls from other traffic generators follow a hop by hop call setup procedure, that is, trying to secure bandwidth on each hop. By contrast, calls that follow the VP contend for bandwidth only at the VP source node. The capacity of the VP varies from 0 to 100 percent of the capacity of the smallest
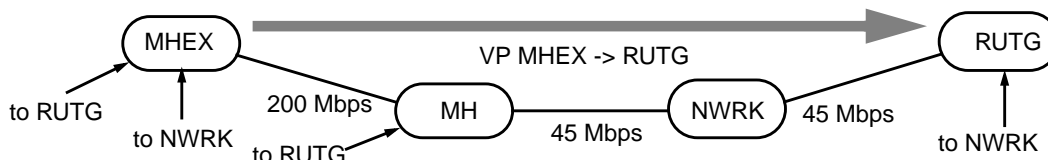
Figure 14: Network Topology for the VP experiment

link on the VP (the MH-NWRK link which is rated at 45 Mbps). When the VP capacity is at 100% only calls from MHEX to RUTG are allowed, since all other calls find no available resources to proceed to their destination. When the VP capacity is reduced to 0, all calls are attempting the regular VC setup procedure. Figure 15 shows the obtained measurements. The throughput curve reveals that
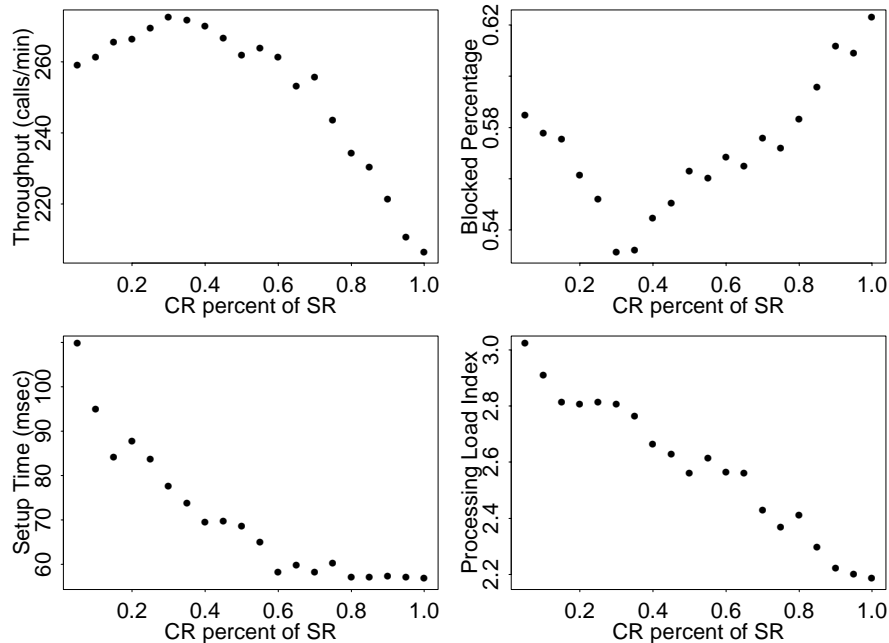

Figure 15: Virtual Path performance vs. allocated networking capacity

the maximum throughput is attained when the VP contract region is approximately 30 percent of the link schedulable region. This happens because below that value, an increasing amount of call requests from MHEX to RUTG find the VP full and use the regular VC setup procedure, thereby forcing the signaling entity at MH and NWRK into an overload state, that causes high call setup times and higher blocking. When the VP contract region increases above 30 percent, the throughput drops slowly as the extra capacity allocated for the VP is partially unused, and as a result a larger percentage of the interfering traffic (that does not follow the VP) is blocked. The fourth plot depicts the average number of signaling messages needed to establish (or reject) an incoming call. The numbers drop as the VP increases in capacity, as calls from MHEX to RUTG follow the VP and use less hops to reach the destination.

4.3 The relation between the Signaling and the Transport Network

The experiments on Xunet have revealed that the signaling network acts as a bottleneck for the transport network. In a commercial environment, a poorly designed signaling system can cause loss of revenue and increased operating costs for the network operator. Researchers have so far disregarded the existence of the signaling system from the overall resource management problem in ATM networks. In reality, users contend for signaling resources *before* contending for networking resources,
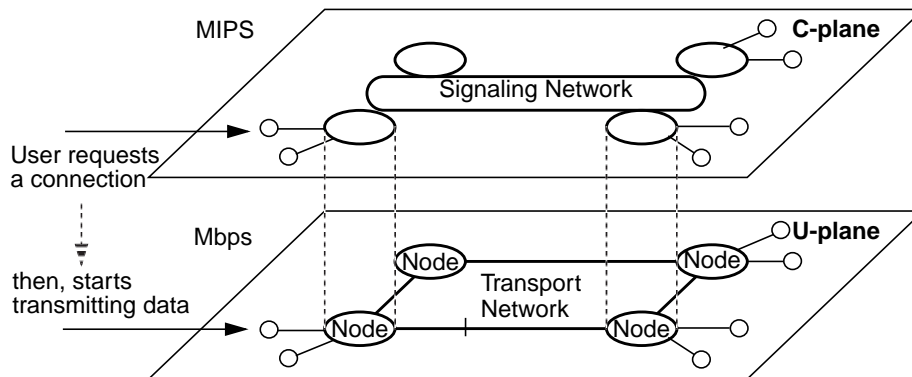

Figure 16: Relation between the Signaling and the Transport Network

since every call is established through a signaling network (Figure 16). In contrast with the transport network which is characterized by networking capacity (e.g., Mbits), the signaling network is a network of computers, and is characterized by processing power (e.g., MIPS). On one extreme, users can generate a large volume of calls that occupy few networking resources. In this situation, the pressure is applied to the signaling network. On the other extreme, users can generate a very low volume of calls that occupy a significant amount of capacity in the network. In this case, it is possible to reach the network capacity without imposing a significant load on the signaling system.

Figure 17 shows a quantitative interpretation of the impact of the signaling system on network throughput. If the signaling system were not present, we would observe the "ideal" behavior: by increasing the offered load, the network throughput would also increase linearly, until it reached the capacity of the network, where it would level-off. By using the results obtained from the Xunet testbed (Figure 13), we can argue that the existence of a signaling system will cause the throughput curve to break at point *B*. The quantity *e* represents the average occupied bandwidth per call. When *e* is small, a larger amount of calls are needed to occupy the same capacity in the network. For small *e*'s the knee of the throughput curve would happen at point *A*, since the signaling network would congest faster. If *e* is larger, i.e., more wideband calls are offered to the network, the knee will appear later at, e.g., point *C*, or might not appear at all.

In the previous cases, the existence of a signaling network breaks the throughput curve before the network capacity is reached. In the Xunet experiments, the congestion of the signaling system becomes so great that no calls can be established and the throughput drops to zero (Figure 13). In reality however, most signaling networks have embedded congestion controls. The role of congestion control is to block a percentage of calls depending on the level of congestion of the signaling system in order to avoid further congestion. With a congestion control mechanism, the network throughput
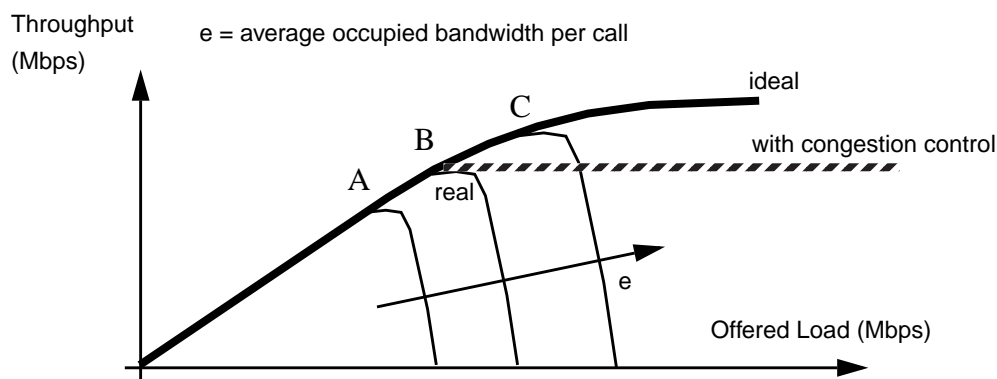
Figure 17: Throughput vs. Offered Load

curve would level-off at the knee point (where the signaling network becomes congested), since any additional load to the network would be blocked. This behavior is still below the ideal throughput that could be achieved from the network.

By using VPs, we can effectively bypass the congestion regions of the signaling system and attain a performance closer to the ideal throughput. A strategic placement of VPs in the network would remove the strain imposed by the VC connection requests from the most loaded signaling processors in the network and possibly enable the network operator to attain a higher throughput. In a network topology like the one of Xunet, signaling congestion first appears in the switches at the center of the network. By using VPs to bypass the central nodes, we were able to more than double the throughput of the network. The question that arises is how to compute the VP distribution policy. The work in [ANE96] presents a complementary framework that optimizes the VP distribution policy with respect to network revenue under Quality of Service constraints.

5. Summary

This paper examined the integration of service and performance management within the Integrated Reference Model. We presented a basic model for managing VC and VP connection services in ATM networks. A set of managed object classes following the OSI standard for network management with complete attribute structure was proposed. The model enables the network manager to retrieve information from agents installed in the network, make decisions based on service configuration and performance observations, and apply a set of controls if the observed performance can be improved. These controls include setting the operating parameters of the signaling system and changing entirely or in part the VP distribution in the network.

Our model was fully implemented on the Xunet ATM testbed. The manager was able to observe the call level performance of Xunet from a dedicated management tool. We presented unique measurements of the call level performance of Xunet and demonstrated the behavior of the network when VPs are in use. These measurements revealed that the signaling system can act as a bottleneck on network thoughpout due to the finite processing resources available to signaling processors. In some cases, calls can be blocked even if capacity is still available in the network. A careful placement of VPs can alleviate most signaling system bottlenecks.

References

[ANE96]   N.G. Anerousis and A.A. Lazar, "Virtual Path Control for ATM Networks with Quality of Service Guarantees", *Proceedings of the 1996 INFOCOM*, San Francisco, CA, March 1996.

[ANE95]   N.G. Anerousis, A.A. Lazar and D.E. Pendarakis, "Taming Xunet III", *ACM Computer Communications Review*, Vol. 25, No. 3, pp. 44-65, October 1995.

[ANE93]   N.G. Anerousis, C.R. Kalmanek and V.E. Kelly, "Implementing OSI Management Facilities on the Xunet ATM Platform," *Proceedings of the Fourth IFIP/IEEE International Workshop on Distributed Systems: Operations and Management*, Long Branch, New Jersey, October 1993.

[BEL91]   J. Bellamy, "Digital Telephony", second edition, John Wiley and Sons, 1991.

[FRA92]   A.G. Fraser, C.R. Kalmanek, A.E. Kaplan, W.T. Marshall and R.C. Restrick, "Xunet 2: A Nationwide Testbed in High-Speed Networking," *Proceedings of the IEEE INFOCOM'92*, Florence, Italy, May 1992.

[GRI95]   D.P. Griffin and P. Georgatsos, "A TMN System for VPC and Routing Management in ATM Networks", in *Proceedings of the 1995 IFIP/IEEE International Symposium on Integrated Network Management*, Santa Barbara, CA, May 1-5, 1995.

[HYM91]   J.M. Hyman, A.A. Lazar, and G. Pacifici, "Real-time scheduling with quality of service constraints,'" *IEEE Journal on Selected Areas in Communications*, vol. 9, pp. 1052-1063, September 1991.

[HYM93a] J.M. Hyman, A.A. Lazar, and G. Pacifici, "A separation principle between scheduling and admission control for broadband switching," *IEEE Journal on Selected Areas in Communications*, vol.11, pp. 605-616, May 1993.

[HYM93b] J.M. Hyman, A.A. Lazar, and G. Pacifici, "Modeling VC, VP and VN Bandwidth Assignment Strategies in Broadband Networks", *Proceedings of the Workshop on Network and Operating Systems Support for Digital Audio and Video*, Lancaster, United Kingdom, November 3-5, 1993, pp. 99-110.

[ICM93]   ICM Consortium, "Revised TMN Architecture, Functions and Case Studies", ICM Deliverable 5, 30 September 1993.

[IET94]   Internet Engineering Task Force, "Definition of Managed Objects for ATM Management", Internet Draft Version 7.0, March 9, 1994.

[ISO92]   Information Processing Systems - Open Systems Interconnection, "Systems Management - Performance Management - Part 11: Workload Monitoring Function", April 1992. International Standard 10164-11.

[ISO91a] ISO, "Information Processing Systems - Open Systems Interconnection: Systems Management - Fault Management - Part 5: Event Report Management Function," July 1991. International Standard 10164-5.

[ISO91b]   ISO, "Information Processing Systems - Open Systems Interconnection: Structure of Management Information - Part 4: Guidelines for Definition of Managed Objects", International Standard 10165-4, July 1991.

[ISO88]   Information Processing Systems - Open Systems Interconnection, "OSI Management Framework," October 1988. International Standard 7498-4.

[KNI91]   G. Pavlou, G. Knight and S. Walton, "Experience of Implementing OSI Management Facilities," *Integrated Network Management, II* (I. Krishnan and W. Zimmer, editors), pp. 259–270, North Holland, 1991.

[LAZ93]   A.A. Lazar and R. Stadler, "On Reducing the Complexity of Management and Control in Future Broadband Networks", *Proceedings of the Fourth IFIP/IEEE International Workshop on Distributed Systems: Operations and Management*, Long Branch, New Jersey, October 1993.

[LAZ92]   A.A. Lazar, "A Real-Time Management, Control and Information Transport Architecture for Broadband Networks", in *Proceedings of the 1992 International Zurich Seminar on Digital Communications*, Zurich, Switzerland, March 1992.

[NEU93]   B. Neumair, "Modeling Resources for Integrated Performance Management", in *Proceedings of the IFIP TC6/WG6.6 Third International Symposium on Integrated Network Management*, San Francisco, California, 18-23 April, 1993.

[OHT93]   S. Ohta and N. Fujii, "Applying OSI System Management Standards to Virtual Path Testing in ATM Networks", *Proceedings of the IFIP TC6/WG6.6 Third International Symposium on Integrated Network Management*, San Francisco, California, 18-23 April, 1993.

[SAR93]   H. Saran, S. Keshav, C.R. Kalmanek and S.P. Morgan, "A Scheduling Discipline and Admission Control Policy for Xunet 2", *Proceedings of the Workshop on Network and Operating Systems Support for Digital Audio and Video*, Lancaster, United Kingdom, November 3-5, 1993.

[TSU92]   M. Tsuchida, A.A. Lazar and N.G. Anerousis, "Structural Representation of Management and Control Information in Broadband Networks", *Proceedings of the 1992 IEEE International Conference on Communications*, Chicago IL., June 1992.

Nikolaos Anerousis received the Diploma in Electrical Engineering from the National Technical University of Athens, Greece, in May 1990, and the M.S., M.Phil. and Ph.D. degrees in Electrical Engineering from Columbia University, New York, NY in 1991, 1994 and 1995. Since 1996 he is a Senior Technical Staff Member in the Networking Research Department of AT&T Research. His interests are in the field of computer and communication networks with emphasis on management architectures for large-scale broadband networks, network performance optimization and service creation. He is a member of the IEEE and the ACM and a member of the Technical Chamber of Greece.

Aurel A. Lazar is a Professor of Electrical Engineering and the Director of the Multimedia Networking Laboratory of the Center for Telecommunications Research, at Columbia University in New York. Along with his long-standing interest in network control and management, he is leading investigations into multimedia networking architectures that support interoperable exchange mechanisms for interactive and on demand multimedia applications with quality of service requirements.

A Fellow of IEEE, Professor Lazar is an editor of the ACM Multimedia Systems, past area editor for Network Management and Control of the IEEE Transactions on Communications, member of the editorial board of Telecommunication Systems and editor of the Springer monograph series on Telecommunication Networks and Computer Systems. His home page address is http://www.ctr.columbia.edu/~aurel.