16

An architecture for performance management of multimedia networks

Giovanni Pacifici and Rolf Stadler
Center for Telecommunications Research – Columbia University
Room 801 Schapiro Research Building
New York, NY 10027–6699
giovanni@ctr.columbia.edu, rolf@ctr.columbia.edu

Abstract

A principal requirement for multimedia networks is the ability to allocate resources to network services with different quality–of–service demands. The objectives of achieving efficient resource utilization, providing quality–of–service guarantees, and adapting to changes in traffic statistics make performance management for multimedia networks a challenging endeavor. In this paper, we address the following questions: what is the respective role of the real–time control system, the performance management system, and the network operator, and how do they interact in order to achieve performance management objectives? We introduce an architecture for performance management, which is based on the idea of controlling network performance by tuning the resource control tasks in the traffic control system. The architecture is built around the L–E model, a generic system–level abstraction of a resource control task. We use a cockpit metaphor to explain how a network operator interacts with the management system while pursuing management objectives.

Keywords

Multimedia networks, performance management, quality–of–service, resource control, network architectures

1 INTRODUCTION

Future multimedia networks will carry traffic of different classes, such as video, voice, and data. Each one of these has its own set of traffic characteristics and performance requirements. Sufficient resources, such as link bandwidth and buffer space, must be allocated to each call of a traffic class in order to guarantee the required quality–of–service (QOS).

As opposed to data networks, which perform best–effort data delivery, the concepts of time and resource are crucial to multimedia networks. Since multimedia networks provide QOS guarantees to user traffic, they contain real–time control functions as part of their traffic control systems. A typical service requirement for a data network is error correction, which is achieved by an end–to–end protocol; a typical requirement for a multimedia network is the guarantee of maximum end–to–end delay on a virtual circuit, which is based on the cooperation of distributed real–time control tasks. Therefore, *the tasks of controlling and allocating resources under QOS constraints*

A. S. Sethi et al. (eds.), *Integrated Network Management IV*

are central in multimedia networks. Note that resources are allocated on various levels of abstraction or granularity, such as per cell, call, or traffic class.

In a multimedia network environment, three entities are involved in the task of controlling and allocating resources — namely, the traffic control system, the performance management system, and the network operator. So far, little work has been done to define the role of these entities and to specify their interactions.

In this paper, we define the task of performance management for multimedia networks and provide an architecture for achieving this task. Specifically, we describe the role of the traffic control system, the performance management system, and the network operator, as well as their interactions. Further, we show how such an architecture relates to a standard management framework like that of ISO/CCITT (ISO, 1991). Two main directions of research activity concentrate on performance management. One direction deals with developing algorithms for resource control tasks that are designed to operate in real–time and make efficient use of resources in a dynamic environment. Usually, these efforts focus on improving the performance of a specific resource control task such as scheduling, buffer management, or admission control. The work described in (Lee and Ray, 1993) is an example of research in this field. The second direction involves activities within the standardized frameworks for network management, such as these developed jointly by the ISO and CCITT committees (ISO, 1991), or by the Internet community (Case et al., 1990; Rose and McCloghrie, 1990). These frameworks provide models to define the structure of management information, and they specify protocols for exchanging this data between functional entities known as managers and agents. Unified modeling of performance–related management information (Neumair, 1993) and the definition of generic interfaces for monitoring (Hayes, 1993) fall into this category.

While recognizing the importance and necessity of the above activities, we follow a third avenue of investigation in this paper, which is essential to meeting the challenges presented by the comprehensive performance management of future multimedia networks. First, our direction focuses on *managing the complete set of resource control tasks* in the traffic control system, by defining a generic abstraction of these tasks. This allows us, from a resource control perspective, to perceive the traffic control system as a collection of resource control subsystems with identical structures and control interfaces. This approach reduces the complexity of the performance management system which controls those subsystems, thus simplifying the design of a performance management framework. Second, having recognized that *performance management attempts to pursue potentially conflicting objectives,* such as the guarantee of QOS versus the obtaining of a high degree of multiplexing, we believe that a system which supports a human operator in implementing the desired strategy is crucial to a performance management framework.

We study *functional descriptions* of a performance management architecture in the form of data flow diagrams. We argue that this kind of description is necessary, in addition to the structural description supported by the standard management frameworks.

The paper is structured as follows. In Sec. 2, we discuss the task of performance management for multimedia networks and outline an architecture to perform this task. Specifically, we define the roles of traffic control and management systems, as well as that of the human operator. In Sec. 3, we refine the architecture by presenting a generic model for resource control tasks and by describing the interaction between the entities involved in the performance management task. Also, we discuss how our architecture relates to the ISO/CCITT management framework. Finally, in Sec. 4, important results of this work are summarized and a few remaining issues are discussed.

2 PERFORMANCE MANAGEMENT FOR MULTIMEDIA NETWORKS

We define the task of performance management for multimedia networks as that of pursuing (high–level) *management objectives.* These objectives can be grouped into two classes. The first class deals with providing network services that meet the needs of customer applications, such as service reliability and QOS guarantees. The second class deals with defining resource allocation strategies that provide benefits for the service provider. Controlling end–to–end packet delays and call blocking rates fall into the first class of management objectives, whereas pursuing high resource utilization and favoring one type of traffic (service) over others fall into the second category. The first class of management objectives favors increasing the resources allocated to each call, while the second class focuses on achieving a high level of resource utilization. These are conflicting requirements, which have to be balanced.

In multimedia networks two different subsystems operate on network resources — namely, a management system and a real–time traffic control system (Lazar, 1991). The following questions arise: What is the role of these systems in the performance management task? How do they interact to achieve high–level management objectives? What is the role of the network operator?

To address these questions, we introduce the architecture outlined in Fig. 1, which contains two subsystems and assumes the presence of an operator. The traffic control system directly regulates the competition for network resources and operates in real–time. The performance management system controls the operations of the traffic control system, while the network operator supervises these activities, pursuing management objectives. The different subsystems in Fig. 1 interact asynchronously and run on different time scales. In order to cope with the high–speed and dynamic nature of user traffic, the real–time traffic control system works on a time scale of μs to ms, while the performance management system and the network operator act on a time scale of seconds or minutes.

In the remainder of this paper, the term "performance management" will refer to the combined activity of all entities in the architecture shown in Fig. 1, whereas the term "performance management system" will be used only for a subsystem of this architecture, and may be thought of as a system structured according to the ISO/CCITT management framework.

2.1 The role of real–time control and performance management

Given the dynamic nature of traffic patterns in a multimedia network, a real–time *traffic control system* is required to regulate the competition for resources among the different traffic classes. The task of this system is to provide the QOS to network users, by utilizing network resources in an efficient way. The traffic control system can be seen as a collection of mechanisms, each of which operates asynchronously and solves a specific resource control problem. Examples of real–time control mechanisms are buffer management and scheduling, flow control, routing and admission control (Gilbert et al., 1991). The operations of the traffic control system can be tuned by changing *control parameters* associated with each mechanism. Changing the parameters of a single controller results in a different resource control policy for that controller and, in turn, may result in a different operating point for all other controllers. The *network state* is the result of the interaction of these real–time control mechanisms.

The task of the *performance management system* is to provide the functionality for pursuing management objectives. The performance management system executes its task by interacting with the real–time control system, following the monitor/control paradigm. This means that it monitors the network state and takes control actions in order to influence this state. Control actions result in changing specific parameters in the real–time control system. The interaction of the performance

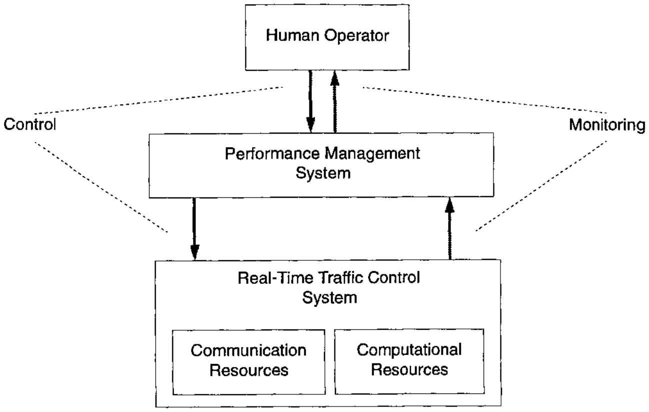

Figure 1: Performance management for multimedia networks

management system with the real–time control system is asynchronous, due to the different time scales on which the functional components in both systems run (Lazar and Stadler, 1993).

The management system is controlled by a human *operator*. Network operators perform actions to influence the network state, and are responsible for achieving management objectives. They monitor the network state represented as dynamic visual abstractions on a graphical interface, and perform operations by acting upon *management parameters*. A detailed example, describing the management parameters used for controlling the traffic mix in a multimedia network, is presented in (Pacifici and Stadler, 1995).

From the above discussion, we gather that the focus of performance management for future multimedia networks is different from that of classical approaches proposed for data networks. Influenced by the OSI Reference Model, performance management is often understood as monitoring and controlling protocol entities and associated service access points (Neumair, 1993; Cellary and Stroinski, 1989). While this is certainly valid for data networks, we argue that, for the case of multimedia networks, the focus should be different — namely, that of managing resource control tasks. In our approach, the performance management system interacts with the real–time control system, which, in turn, operates on protocol engine parameters and network resources. Executing performance management functions means operating management parameters that tune resource control tasks. We justify our point of view by the fact that multimedia networks provide real–time services, and resource control plays a central and critical role. Data networks, such as the existing Internet, do not guarantee QOS, and, as a result, their resource control tasks are much less complex.

2.2 The role of the network operator — the cockpit metaphor

Since the network is the heart of every distributed service, the failure of large parts of a network can result in a disaster for customers, and, as a consequence, for the service provider. Therefore, experienced operators supervise the operation of a network to prevent such scenarios from occurring. As we explained in the last section, supervision for future multimedia networks may be even more important than for today's networks, due to the complex interactions inside the traffic control systems. To explain the role of human operators and the way they interact with the management system while pursuing management objectives, we use the metaphor of a pilot flying an airplane.

A pilot operates the aircraft in reaction to and in anticipation of environmental conditions, as expressed by wind, visibility, air pressure, etc. The pilot has no influence on the environment and on how it evolves. In a similar way, a network operator performs actions to handle the current and anticipated load pattern of the network traffic, while guaranteeing the required QOS to network services and allowing a high utilization of network resources. The traffic load pattern changes over time and cannot be influenced by the operator. However, operators are responsible for maintaining the network state within a stability region that allows reliable operations. When the traffic pattern changes, so does the network state, and the operator "navigates" the network state back into the stability region, if necessary.

A pilot operates on high–level controls such as yoke, handles, and control sticks, the positions of which relate to specific settings of the airplane's control surfaces such as elevators, ailerons, rudders, and flap positions. Similarly, the network operator sets management parameters. Modifications to these parameters are translated by the management system into control parameters that influence the way network control mechanisms operate, thereby affecting the network state. Operators observe the reaction of the system in response to control actions in the same way a pilot observes the flight instruments changing to adjustments of the flight controls. The relationships between an aircraft's speed and vertical velocity, on the one hand, and elevators and throttle, on the other, are complex, and a pilot understands them through practice. Likewise, we think that understanding certain relationships between management parameters and the network state in large multimedia networks will be based in large part on experience and expertise.

While steady–state conditions hold, an autopilot system can control the aircraft and perform automated functions. In difficult situations or during unprecedented events, however, the pilot takes control. Such situations might include a sudden change in the weather or the occurrence of turbulences. Also, the takeoff and landing procedures are normally executed by the pilots themselves. We believe that, in an analogous way, performance management functions can be automated when the network operates in a stability region subject to minor fluctuations in the traffic load patterns. Operators, however, will always be needed to handle difficult situations. In such conditions, they will decide which functions should be executed and when they should be run assisted perhaps by an expert system. Aircraft takeoff and landing operations can be compared to adding or removing parts of the network during operation — tasks that have to be performed in every network on a regular basis and need human supervision.

3 AN ARCHITECTURE FOR PERFORMANCE MANAGEMENT

In this section, we develop a performance management architecture that integrates the network subsystems that participate in the resource management task. We present an abstraction of the traffic control system with respect to resource control and utilize this model to define a framework that allows management operations to influence the behavior of traffic control mechanisms.

3.1 Modeling resource control tasks — the L–E model

The traffic control system of multimedia networks contains a collection of resource control subsystems, each of which implements a specific task, such as admission control or routing. Each of these subsystem regulates access to a specific resource, by responding to requests that are generated by functions external to the resource control subsystem. The behavior of the resource control task (i.e., the way it responses to service requests) can be influenced by changing a set of control parameters associated with the subsystem.

The main functional components, of a resource control subsystem, together with the interactions among components and with the outside world, are identified in the *L–E model* shown in Fig. 2. We use a functional model in Fig. 2, in order to focus on functional components as well as the data exchanged and accessed by them (Rumbaugh et al., 1991).

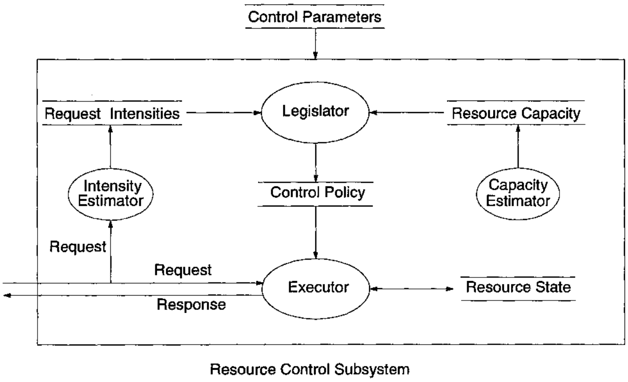

Figure 2: Data–flow diagram of the L–E model

The main idea behind the L–E model is that the task of computing a control policy for allocating a resource in a dynamic environment is separated from the task of binding this resource to a particular communication service. Following this separation, the model contains two types of mechanisms, the *legislator* and the *executor* (see Fig. 2). A pair of these mechanisms, one of each type, interact to perform a specific resource control task, e.g., controlling access to a physical network link.

The *legislator* generates a set of rules, which must be observed when allocating a resource. This set of rules is called the *control policy*. The *executor* regulates access to the communication resource while observing the current control policy. In other words, the executor implements the control policy computed by the legislator.

The executor is driven by external stimuli. Its task is to serve requests that are initiated by functions external to the resource control subsystem. The legislator, in contrast, is either invoked by the executor or runs on its own and periodically recomputes the control policy. It performs its operation usually on a time scale much slower than that of the executor, since the computational complexity of a resource control subsystem resides in the legislator part.

Legislator and executor interact by sharing a data object — the control policy — which is written by the legislator and read by the executor. The interaction between legislator and executor can be either synchronous or asynchronous. In the synchronous case, the legislator invokes the executor, e.g., in the form of a function call. The routing scheme in the plaNET traffic control system (Gopal and Guerin, 1994) works in this way. In the case of asynchronous interaction, legislator and executor form a loosely coupled subsystem. Each mechanism runs on its own time scale, and they communicate asynchronously via the shared policy object. This approach can be found in the adaptive routing schemes of today's long distance telephone networks (Girard, 1990). Note that asynchronous interaction between legislator and executor allows them to run independently and on different time scales. Therefore, they can be optimized according to different requirements: the executor guarantees response times, while the legislator optimizes the utilization of the resource, e.g., by minimizing a given cost function.

The L–E model allows for a wide range of possible implementation decisions. It covers single threaded, distributed, as well as parallel implementations of resource control subsystems, depending on whether the mechanisms are intended to run on the same or different machines and whether their interaction is designed to be synchronous or asynchronous. Further, the model supports the case where several executors share the same legislator.

In order to manage resources in an efficient way, the resource control system of multimedia networks must be able to adapt dynamically to changes in the network state and traffic statistics. In the L–E model this is achieved by the legislator, which periodically recomputes the control policy, taking into account the latest value of the request intensities and the resource capacity.

Our model contains two mechanisms that generate the dynamic abstractions needed by the legislator to recompute the control policy. The *intensity estimator* calculates the request intensities, by filtering the stream of service requests, and the *capacity estimator* computes the resource capacity, based on traffic statistics and configuration data. Note that the capacity of a network link (expressed in cell/sec) can be seen as a constant configuration parameter, while the capacity of a high–level abstraction of the same link (i.e, the maximum number of video, voice and data calls that can be multiplexed at any given time on that link) varies continuously, following changes in traffic characteristics. Examples of capacity estimation techniques that provide high–level abstractions of link resources can be found in (Ferrari and Verma, 1990; Hyman et al., 1991). Both the intensity and capacity estimators run on the same time-scale as the legislator and generate new estimates for each new computation of the control policy.

The L–E model provides the framework for dynamically influencing the resource control task, by associating *control parameters* with each mechanism, i.e., with legislator, executor, intensity estimator, and capacity estimator. Control parameters of a legislator include the QOS constraints for handling requests and the utility generated for granting access to the resource, as well as the time interval between two consecutive recomputations of the control policy. The length of the estimation interval, which reflects the capability of the system to respond to changes in the traffic statistics, is a typical control parameter for the intensity estimator. The robustness of the capacity estimation processes is a parameter associated with conflicting objectives. In the case of link admission control, it relates to the trade–off between using the link bandwidth efficiently and providing cell–level QOS guarantees (Pacifici and Stadler, 1995).

All these control parameters provide the fundamental capability to influence how a resource control system works, namely, by affecting the QOS constraints under which it operates, its adaptivity related to changes in the environment, and its robustness in guaranteeing the QOS under varying traffic loads and conditions.

The L–E model is based on our experience with designing and implementing traffic control mechanisms for multiclass networks. Tab. 1 identifies some elements of the L–E model for the most important resource control tasks in a multimedia system. For example, the TCP/IP flow control task (Jacobson, 1988) can be modeled as an end–to–end protocol entity (executor) that performs transport operations according to a maximum window size (control policy). The window size is determined by the flow controller (legislator), which computes the size of the window using the estimated link bandwidth available to a specific user (capacity estimation) and the transmission rate (request intensities) of the specific user source (Jacobson, 1988). The system state is defined by the number of transmitted cells not yet acknowledged.

The tasks of scheduling and buffer management — to give another example — can be modeled in the same fashion. Here, the policy is defined by time sharing (scheduling) and space partitioning

Task	Control Policy	Resource State	Resource Capacity	Request Intensities
Admission Control	State transition matrix	Number of active calls	Schedulable Region	Call arrival rates and call holding times
VC Routing	Set of routes	Number of active calls per link	Collection of Schedulable Regions	Call arrival rates and call holding times
Flow Control	Window size	Number of cells in system	Available link bandwidth	Cell arrival rates
Buffer Mngt.	Buffer partitions	Number of cells in buffer	Buffer space	Cell arrival rates
Scheduling	Link partitions	Number of cells in buffer	Link bandwidth	Cell arrival rates

Table 1: Modeling resource management tasks in a multimedia network

(buffer management) of the resources among each traffic class. The system state is determined by the number of cells in the buffer, while the request intensities are given by the cell arrival and departure rates. The link speed and the buffer size define the resource capacities, which are available as configuration parameters. The admission control task and its functional model are discussed in (Pacifici and Stadler, 1995).

With the above discussion we want to illustrate that that our model is truly generic in the sense that it is not restricted to a particular resource control task. Note that Table 1 is based on specific control algorithms. The choice of different algorithms can result in different table entries for control policy, resource state, etc.

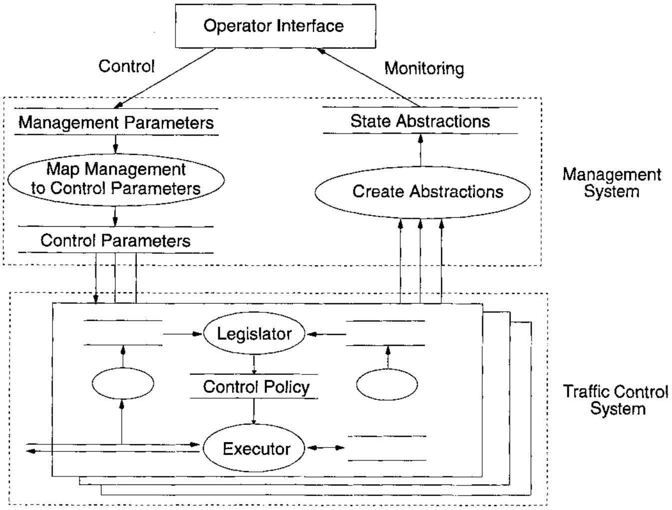

Figure 3: Interaction between the operator, the management system, and traffic control tasks

3.2 Integrating resource control and performance management

From the point of view of performance management, the traffic control system can be seen as a set of subsystems, each performing a specific resource control task. As described in Sec. 3.1, a set of control parameters can be associated with each resource control subsystem. These parameters define the control interface between the management and traffic control systems. The management system writes them while the traffic control system reads them. This scheme allows for asynchronous interaction between functional components of both systems, thus enabling these components to run on different time scales and at different locations. By modifying control parameters, the management system influences the behavior of a resource control subsystem, and, therefore, changes the way resources are allocated.

There are two main reasons for including the L–E model in a framework for performance management. First, in order to tune resource allocation, specific knowledge about the algorithms involved in resource control and the way the resource control subsystem is implemented is not required in the management system. This allows a clear split between performance management and the traffic control system, with the set of control parameters defining the control interface. Second, the L–E model provides generic classes of control parameters that can be made accessible to the management system.

The management system presents a high–level view of the network state to the operator in the form of dynamic visual abstractions. The operator manipulates a set of management parameters. Changes in these parameters are translated into modifications to control parameters that influence the behavior of traffic control components (see Fig. 3).

A straightforward way to support an operator with management capabilities is to make each control parameter directly available at the operator interface. For example, a control parameter that is defined within a certain interval can be associated with a management parameter (both values may be related by a monotonic mapping, such as with a linear or logarithmic function), which can be presented to the operator by means of the visual abstraction of a slider. Changing the position of the slider will result in a change of the control parameter, which, in turn, will affect the corresponding resource control subsystem.

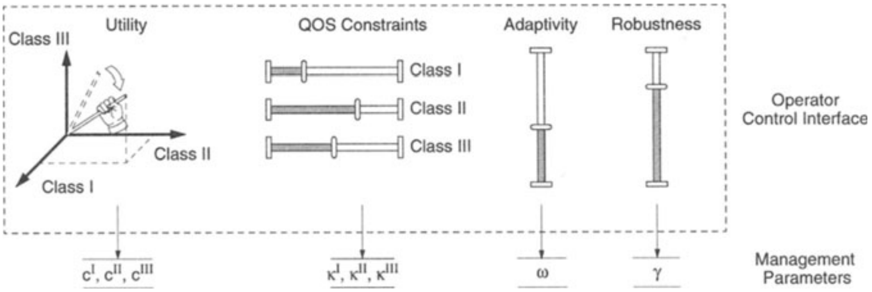

Figure 4: Visual abstractions and management parameters associated with the task of managing the communication resources of a multimedia network

Figure 4 introduces a sample set of management parameters associated with the task of managing the communication resources of a multimedia network, and shows the visual abstractions that allow an operator to change the management parameters, thus affecting the performance of the network. In this example, the management parameters relate to network utility, QOS constraints, as well as adaptivity and robustness of the resource control system. In (Pacifici and Stadler, 1995) it is shown how the task of link admission control can be managed, by using these four different types of management parameters.

Obviously, a network operator needs the capability to tune not only each single controller in the traffic control system, but also sets of controllers simultaneously, for example, all controllers on a specific route or inside a certain network region. Therefore, the operator interface provides *selection capabilities* that allow an operator to choose a set of objects (e.g., links, nodes, network regions, or the whole network) that determine the domain of controllers on which a management operation is to be executed. A management operation thus involves a selection operation and the setting of a management parameter. The management system then maps this data onto both the settings for control parameters and the domain of controllers affected by the operation, and distributes the settings to the traffic control system.

Note that a single management parameter can be associated with several classes of controllers. A management parameter related to robustness, for example, can be associated with control parameters in resource control systems that implement call routing, call admission control, and cell scheduling. Again, the mapping from the management to the various control parameters is performed by the management system.

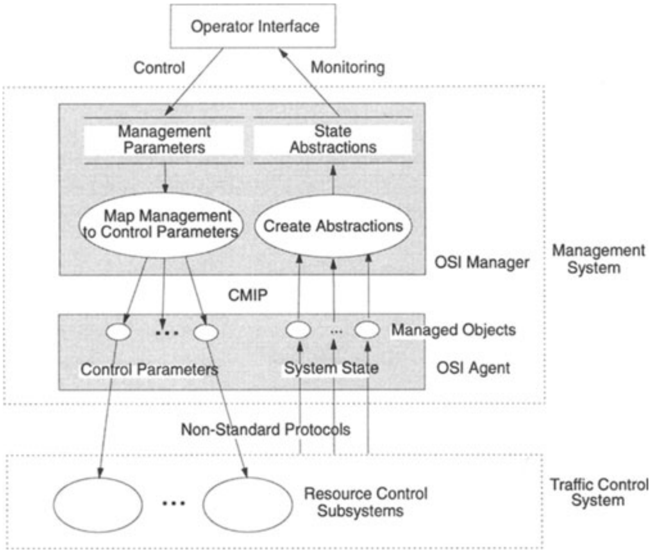

Figure 5: Performance management within the OSI framework

Having described the concepts of our architecture, the question arises, how do they relate to a management framework, such as the one standardized by ISO (ISO, 1991)? In that framework, the system to be managed is conceptualized as a global database, the Management Information Base (MIB). The MIB contains a set of managed objects, which represent network entities. Managed objects are implemented on OSI agents, and can be accessed and manipulated by OSI managers by a standard protocol called CMIP. Therefore, monitoring and controlling a system means reading and changing managed objects in a standardized way.

Figure 5 shows our approach. We propose that the control parameters associated with network mechanisms be modeled and implemented on agents as managed objects, which are part of the management system. Further, network state information should be modeled and implemented in the same way, and thus be accessible for management purposes. The mapping and abstraction functions should be implemented on a manager, because they support network functions that operate on the global space of managed objects, which will be distributed over several agents. While the interaction between the manager and the agents is standardized, there is no standard protocol for the communication between a managed object and a resource control mechanism.

4 DISCUSSION

We believe that the architecture presented in this paper opens the way for building powerful tools for network operators who manage the resources of a multimedia network. The selection functionality allows them to choose a set of objects on the operator interface, so as to define a domain of controllers (such as a link, a path, a network region, or the whole network) on which a management operation is to be executed. Operators can change, for every selected domain, the QOS constraints and the utility generated by the user traffic in this domain, and they can tune the adaptivity and robustness of resource control functions in the same fashion. These tools support network operators in their task of navigating the managed system — here we use a term from the cockpit paradigm — effectively and safely. Operators have at their disposition high–level controls in order to keep the appropriate balance when pursuing different, potentially conflicting objectives. These objectives include providing QOS on the cell–level and call–level, keeping up a high degree of multiplexing, securing network utilization, and maintaining a highly responsive and yet stable system.

We are currently experimenting with the design of our architecture using a network emulator, which runs functional components of a traffic control and management system of a multimedia network. The emulator is implemented on a KSR parallel machine. It emulates a 50 node network, in which traffic statistics can be dynamically changed at every network access point. The operator interface runs on an Indigo2 workstation, which is connected to the KSR via an ATM link. We can demonstrate, for example, how the traffic mix in the network can be influenced by executing management operations that affect link resource controllers in selected network domains. The effect of management operations can be observed in real-time, using the capability of visualizing call blocking rates and network utilization for any selected network domain.

All examples presented in this paper relate to managing communication resources — indeed, one of the classic subjects in traffic control. Since our framework is generic, other resources, such as computational resources, can be included. Because the traffic control system needs resources to operate, these can be abstracted using the L–E model, and, therefore, their performance can be managed according to our framework. For telephone networks, performance management of traffic control systems has been recognized as a crucial issue (Kühn et al., 1994), and we believe that it will play an equally important role in emerging multimedia networks.

Finally, we believe that our framework can be applied to *managing the performance of real–time services*, such as access to a video server or to a multimedia database, since the resource control systems associated with these services can be abstracted using the L–E model. Furthermore, it can be extended to include the computational resources of multimedia workstations, thus leading to a framework for managing and controlling resources in a distributed multimedia application environment. The architecture proposed in (Campbell et al., 1994) can be seen as a step in this direction, though network management aspects are not addressed there. Note that our approach allows the *integration of the network management and service management tasks* — as far as performance is concerned — which opens interesting perspectives for further investigation.

References

Campbell, A., Coulson, G., and Hutchison, D. (1994). A quality of service architecture. *Computer Communication Review*, 24(2):6–27.

Case, J., Fedor, M., Schoffstall, M., and Davin, C. (1990). *A Simple Network Management Protocol (SNMP)*. RFC–1157.

Cellary, W. and Stroinski, M. (1989). A performance management architecture for protocol entity optimization. In Meandzija, I. B. and Westcott, J., editors, *Integrated Network Management, I*, pages 227–234. Elsevier Science (North–Holland), Amsterdam, The Netherlands.

Ferrari, D. and Verma, D. C. (1990). A scheme for real-time channel establishment in wide-area networks. *IEEE Journal on Selected Areas in Communications*, SAC-8(3):368–379.

Gilbert, H., Aboul-Magd, O., and Phung, V. (1991). Developing a cohesive traffic management strategy for ATM networks. *IEEE Communications Magazine*, 20(10):36–45.

Girard, A. (1990). *Routing and Dimensioning in Circuit–Switched Networks*. Addison–Wesley, Reading, MA.

Gopal, I. and Guerin, R. (1994). Network transparency: The plaNET approach. *IEEE/ACM Transactions on Networking*, 2(3):226–239.

Hayes, S. (1993). Analyzing network performance management. *IEEE Communications Magazine*, 31(5):52–58.

Hyman, J. M., Lazar, A. A., and Pacifici, G. (1991). Real-time scheduling with quality of service constraints. *IEEE Journal on Selected Areas in Communications*, 9(7):1052–1063.

ISO (1991). *Information Processing Systems — Open System Interconnection — Systems Management Overview*. ISO/IEC, IS 10040.

Jacobson, V. (1988). Congestion avoidance and control. In *Proceedings of the ACM SIGCOMM*, pages 316–329, Stanford, CA.

Kühn, P. J., Pack, C. D., and Skoog, R. A. (1994). Common channel signaling networks: Past, present, future. *IEEE Journal on Selected Areas in Communications*, 12(3):383–394.

Lazar, A. A. (1991). An architecture for real-time control of broadband networks. In *Proceedings of the IEEE Global Telecommunications Conference*, pages 289–295, Phoenix, AZ.

Lazar, A. A. and Stadler, R. (1993). On reducing the complexity of management and control of broadband networks. In *Proceedings of the Workshop on Distributed Systems: Operations and Management*, Long Beach, NJ.

Lee, S. and Ray, A. (1993). Performance management of multiple access communications networks. *IEEE Journal on Selected Areas in Communications*, 11(9):1426–1437.

Neumair, B. (1993). Modeling resources for integrated performance management. In Hegering, H. and Yemini, Y., editors, *Integrated Network Management, III*, pages 109–121. Elsevier Science (North–Holland), Amsterdam, The Netherlands.

Pacifici, G. and Stadler, R. (1995). Integrating resource control and performance management in multimedia networks. In *Proceedings of the IEEE International Conference on Communications*, Seattle, WA.

Rose, M. and McCloghrie, K. (1990). *Structure and Identification of Management Information for TCP/IP based Internets*. RFC–1155.

Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., and Lorensen, W. (1991). *Object–Oriented Modeling and Design*. Prentice–Hall, Englewood-Cliffs, NJ.

Giovanni Pacifici received the *Laurea* and the *Research Doctorate* degrees from the University of Rome "La Sapienza" in 1984 and 1989 respectively. As a student, his main activities were focused on the performance evaluation of access control protocols for local and metropolitan area networks, with an emphasis on the integration of voice and data. In the course of his studies, he was a Visiting Scholar at the Center for Telecommunications Research, Columbia University, where he designed and implemented a monitoring and traffic generation system for MAGNET II, a high speed metropolitan area network. In 1989, he joined the staff of the Center for Telecommunications Research as a Research Scientist. His research interests include resource control, performance management and real-time quality of service estimation for broadband networks. Dr. Pacifici is a member of IEEE and ACM.

Rolf Stadler received a master's degree in mathematics and a Ph.D. degree in computer science from the University of Zurich in 1984 and 1990, respectively. His thesis work focused on the specification of communication systems. During 1991 he was a post-doctoral researcher at the IBM Zurich Research Laboratory, involved in developing a traffic management system for a broadband LAN/WAN environment. From 1992 to 1994 he was a Visiting Scholar at the Center for Telecommunications Research, Columbia University. In 1994 he joined the staff of the Center for Telecommunications Research as a Research Scientist. His current interests include management, control, and services with respect to broadband networks. Dr. Stadler is a member of IEEE and ACM.