Service and traffic management for IBCN

by K. Geihs
P. Francois
D. Griffin
C. Kaas-Petersen
A. Mann

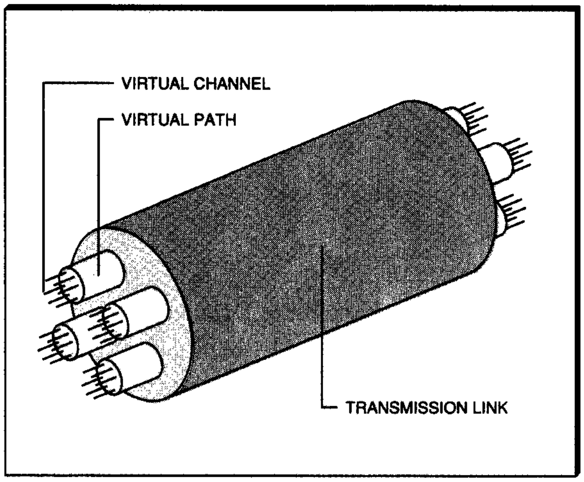
The future Integrated Broadband Communications Network (IBCN) will provide high-speed communication capabilities that support a variety of existing and new services. The management of such a complex environment requires innovative management systems. NEMESYS is a project within the European Commission's Research and Development in Advanced Communications in Europe (RACE) program. The project goals are to demonstrate and evaluate the use of advanced information processing techniques for quality-of-service and traffic management. To reach these goals, a series of experimental prototypes are being built. This paper describes the assumptions, objectives, and approach of NEMESYS, and in particular, the design and implementation of an experiment that investigates service and traffic management techniques in a simulated asynchronous transfer mode environment. Because the project is not yet finished, some preliminary results are presented.

The future Integrated Broadband Communications Network (IBCN) will support all existing services and, together with novel multimedia technologies, will stimulate the emergence of a variety of new services in areas such as education, entertainment, business, and personal communication.[1] Thousands of services are available today on the low-performance, widely accessible videotex (e.g., banking, weather forecast, games, job offers). What will be achievable with the introduction of multimedia technology, such as broadband live video systems, stretches our imagination.

New transmission technologies and the integration of existing and new services will pose new management requirements for the introduction of IBCN. In the envisaged environment, a service is provided by a cascade of underlying networks and subservices that will be offered by a variety of providers, some very large and some very specialized. All of them will need to cooperate in management functions such as billing, performance, quality of service, and troubleshooting, so that the "last" service provider facing the "final" end user is able to deliver the desired quality of service. Additionally, the general trend toward fair competition (e.g., the Open Network Provision directive in Europe) and growing customer requirements are leading to the definition of precise contractual quality of service. Therefore, network and service providers need powerful tools allowing them to measure how they meet their commitments and to optimize the usage of their systems.

Research and standardization is well under way to provide the foundations for the IBCN. The proposed broadband packet technology, i.e., asynchronous transfer mode (ATM),[1] which is based on statistical multiplexing, allows the optimiza-

Figure 1 Asynchronous transfer mode (ATM) links, virtual paths, and virtual channels



tion of network resource utilization and is very well fitted to the transmission of heterogeneous traffic (voice, video, data) with highly dispersed characteristics (e.g., transmission rate, ratio of peak to mean bandwidth, maximum delay, tolerable error rates). However, the algorithms and methods related to this multiplexing are complex and need further studies. A key problem is to guarantee a committed quality of service while maintaining a high utilization of network resources. The customer, service provider, and network provider may have conflicting objectives. Today's telephone network with fixed reserved bandwidth and the packet data networks with fairly homogenous traffic do not provide comparable experiences.

The European Research and Development in Advanced Communications in Europe (RACE) initiative is funding research into the technological foundations for the commercial introduction of the IBCN. Adequate management solutions are considered to be a very important prerequisite for the operation of the IBCN. RACE defines management as the end-to-end management of networks and services offered to the customers. Facing the above problems, there is a need to prototype IBCN management functions very early, including the architecture and standards defined, and to experiment with the latest programming techniques that could help to master the enormous task of management development.

Project NEMESYS addresses several research areas. The main focus is on the application of advanced information processing techniques, such as expert systems, neural networks, distributed systems, and object-oriented design and implementation, to the critical problem of traffic management and quality-of-service management in an IBCN environment. Further topics include the architectural structuring of interacting but otherwise independent organizations, such as network providers, service providers, and customers, and the application of management standards. The NEMESYS consortium is developing three consecutive prototypes over a five-year period. This paper reports the status of the project after the first four years, during which two prototypes have been built, and highlights the design and implementation of the third prototype to be completed by the end of 1992. Preliminary results are presented.

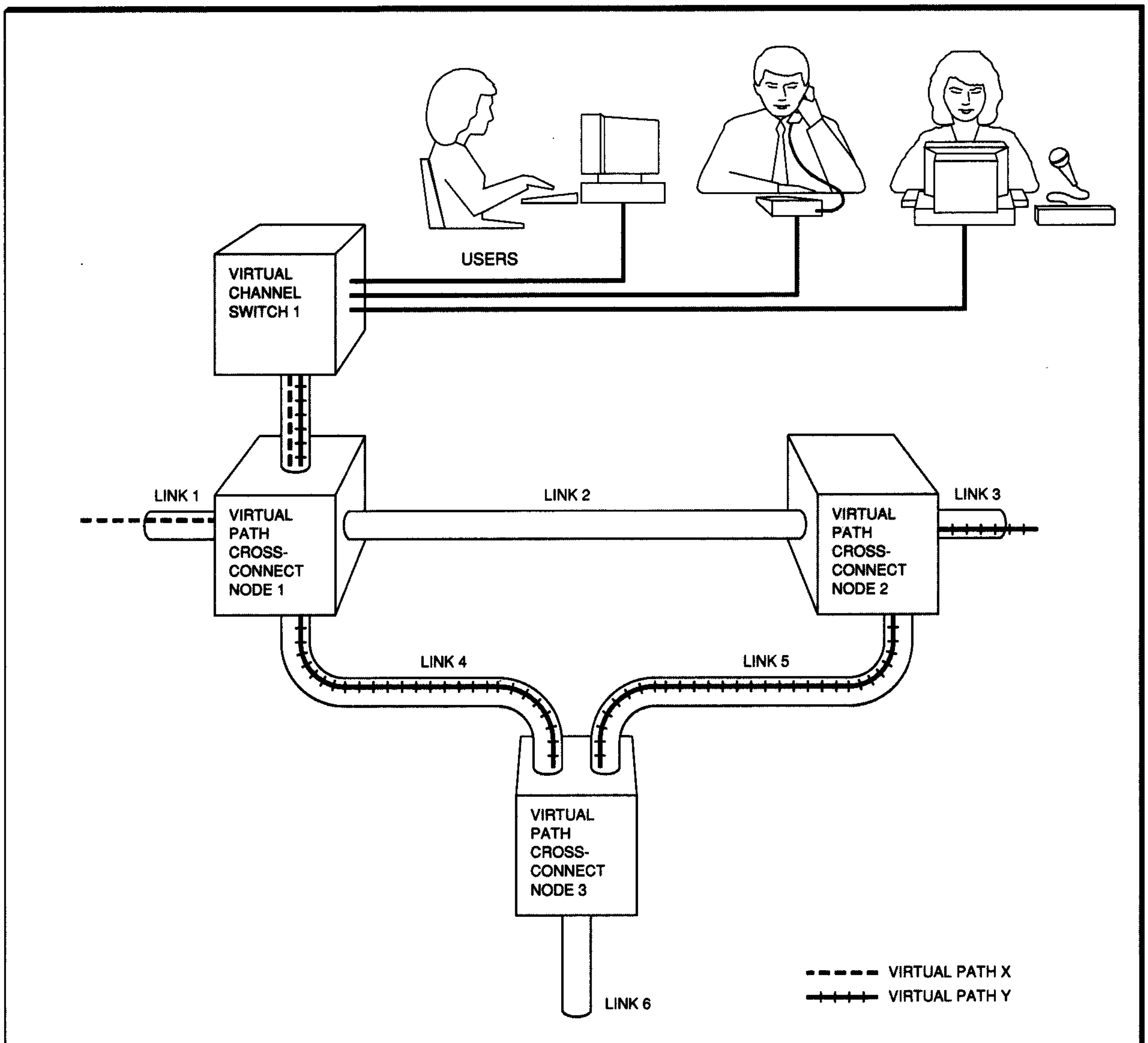
Future service environment

On the way toward implementing the IBCN, the broadband ISDN (Integrated Services Digital Network) is considered to be an evolutionary step that augments the existing narrowband ISDN with broadband services such as video-telephony, video-conferencing, high-speed file transfer, and multimedia document transfer.[1] ATM has been selected as the switching technique for the broadband ISDN.[1]

In ATM networks all data are transferred in fixed-size minipackets called "cells." An ATM cell consists of a 5-octet header and a 48-octet information field. Cells are identified on a link by their header, which contains a virtual channel identifier and virtual path identifier. Cells belonging to a particular connection are allocated to a virtual channel, and groups of virtual channels are allocated to virtual paths, which in turn are grouped onto a link (see Figure 1).[2]

Virtual paths will usually span more than one link, and cells belonging to a particular virtual path will be routed between links at virtual path cross-connect equipment. (See the virtual path y in Figure 2.) Cells belonging to particular virtual channels can be switched between virtual paths at virtual channel switches. Figure 2 shows virtual path x on link 1 and terminating at virtual channel switch 1, and virtual path y starting at node 1 and being routed over links 4, 5, and 3. Cells required to

Figure 2 Asynchronous transfer mode (ATM) routing and switching functions



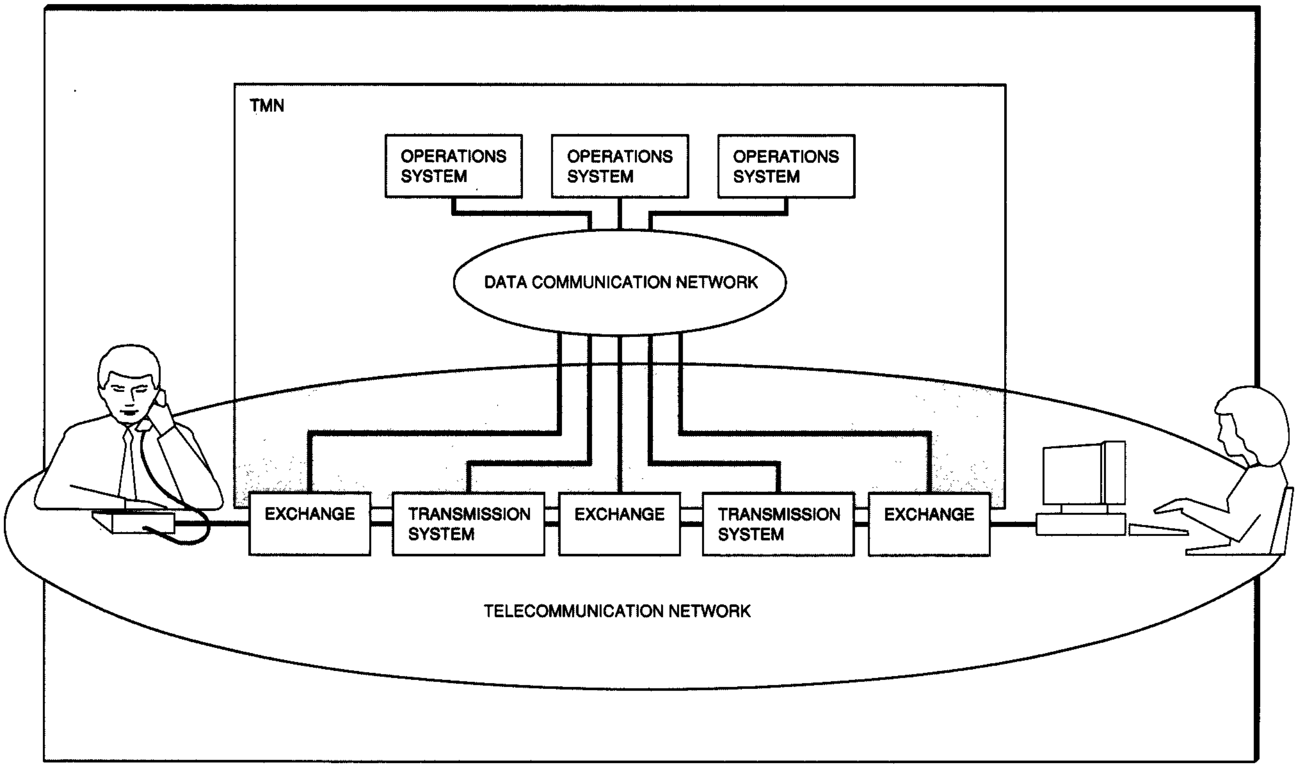
traverse virtual paths x and y will be switched at virtual channel switch 1.

Note that although there are two distinct routing/switching functions, it may be the case that, in a particular implementation, virtual path cross-connects and virtual channel switches are all physically located in a generic ATM switch.

The NEMESYS consortium has assumed an ATM communication environment and service types as envisaged for the broadband ISDN. The assumptions involved are described in a later section on service management.

Services are classified by the International Telegraph and Telephone Consultative Committee (CCITT) as bearer services, teleservices, and supplementary services. These services in the IBCN may be provided and operated by independent organizations. Some basic definitions and model assumptions can be found in Reference 3. A *ser-*

Figure 3 Telecommunications Management Network (TMN) and IBCN



vice is a set of functions offered to a user by an organization. Such an organization is called the *service provider.* The *service user* is a customer to the service provider. Proper operation of a service is intimately tied to the notion of *quality of service* (QoS). QoS is defined to be the collective effect of service performances that determines the degree of satisfaction of a user of the service. Managing the network environment in order to ensure desired levels of quality of service is a central task of the management system.

According to recommendation M.3010[4] of the CCITT, the management issues of a telecommunication network are conceptually separated in the Telecommunications Management Network (TMN). A TMN supports the management requirements of administrations to "plan, provision, install, maintain, operate and administer telecommunications networks and services."[4] The separation of management and telecommunication functions is illustrated in Figure 3. The figure shows that the TMN functionality is provided in a distributed fashion and that it uses a communication network that is (at least logically) separate from the telecommunication network. The managed equipment (e.g., exchanges and transmission systems) of the telecommunication network is called network elements. The operations system components perform application functions such as service and traffic management. More details on the TMN functional components and their various interfaces may be found in recommendation M.3010.[4]

TMN uses a layered management model. Each layer performs a set of logically related management functions. One example set of functional layers[4] is: business management, service management, network management, and network element management.

The business management layer contains functions related to the business obligations and policies of the enterprise. It interacts with other enterprise management functions. The service management layer ensures the contractual aspects between service providers and service users, e.g., agreements on a certain quality of service. The network management layer is responsible for managing all the network elements at a global or domain scope. The network element management layer manages individual network elements and provides an abstraction of the physical resources in the network. Adjacent layers interact frequently in order to achieve their goals.

The TMN information model is based on an object-oriented model as described in the Open Systems Interconnection (OSI) management standards.[5] Manageable physical or logical resources are modeled as *managed objects* represented by an *agent* that interacts with the *manager* or *managers* who control the managed object or objects. Each functional layer contains a number of such managed objects and may act as manager/agent for the lower or higher layer, respectively.

NEMESYS approach

The primary objective of the NEMESYS project is to demonstrate and evaluate the use of advanced information processing (AIP) techniques for the implementation of network management applications in the area of traffic and quality-of-service (QoS) management for IBCN. AIP is a term from the European Strategic Programme for Research and Development in Information Technology (ESPRIT) and RACE programs and is used to describe state-of-the-art information processing techniques. Traffic and QoS management is part of performance management and has been defined by the NEMESYS project as "the operational and administrative facilities required to optimize the utilization of network resources whilst maintaining QoS as received by the users of the network."[6]

Because the project is investigating the application of AIPs to traffic and QoS management, it would not be useful to attempt to evaluate AIP techniques in isolation from the management system—they must be evaluated in the context of the particular management functions. Demonstration of the employed AIP techniques is being accomplished by the construction of experimental prototypes using specific state-of-the-art information processing techniques and designed to address a specific subset of traffic and QoS management functions. However, evaluation of the AIP techniques is more complex, and reconciling the different perspectives of the telecommunication systems designer and the software developer on the scope of an evaluation adds to the problem. In general, the demonstration places more emphasis on the nonfunctional requirements, whereas the evaluation places more emphasis on a subjective assessment of the tools used. Both cases assume that the primary functional requirements are satisfactorily met.

QoS is the collective effect of service performances that determines the satisfaction of a user of the service.

For example, consider the evaluation of an AIP such as rule-based systems for making decisions on whether new calls should be accepted onto a network. The telecommunications systems designer is interested in issues such as the speed of the system and the implications on the hardware necessary to run the code. The software developer looks at aspects such as the maturity of documentation, support for the tool, and the maintainability of the code. Both groups would not even consider using the AIP if it were not possible to develop a system that makes correct decisions on whether new calls should be accepted or not.

In order to structure the evaluation process and to resolve some of the differences of emphasis, an AIP evaluation methodology was created to elaborate a set of evaluation criteria and to measure the performance of the management system, and hence the AIP techniques used to implement the systems against those criteria. The first step of the methodology is to determine the functional and nonfunctional requirements of the management system and its constituent components. Some functional requirements are network efficiency gain, ability to adapt to network usage changes, and QoS received by users. Some nonfunctional requirements are real-time response, processing requirements, and storage requirements.

The second step is to derive the requirements on AIP techniques from the system requirements together with any additional requirements from the point of view of the developer, such as reusability, ability to be integrated with other techniques or tools, and productivity.

The list of requirements from the first two steps is then used to create a set of evaluation criteria—at least one criterion for each requirement. For example: Functional requirements—Does the system generate a route from each access node to every other node? Nonfunctional requirements—What is the time taken to generate a routing table? Developer requirements—How long does it take to develop the code to generate the routing tables? What is the availability of computer-assisted software engineering (CASE) tools for this AIP technique?

Candidate AIPs are selected and a prototype management system is developed using these techniques. Where possible, more than one technique, conventional and AIP, is used to implement the same management function. A series of experiments are then carried out on the resulting system to make measurements against the evaluation criteria created earlier. Comparisons can then be made between the various implementation options to show the performance of different techniques against the criteria. At the same time the implementers are questioned, to gain an assessment of the techniques from the developer's point of view.

The project's five-year lifetime has been divided into three experimental cycles, with the intention that each experiment will build upon the knowledge gained during the previous one. NEMESYS began with three in-depth studies. The first study concerned the state of the art of information processing, reviewing the latest research areas and the availability of tools. The second study concerned the requirements of traffic and QoS management for IBCN, including a review of the detail of ATM networks, an understanding of the nature of multiservice networks, and the possibilities of traffic and QoS management in this scenario. The third study was on the emerging standards for network management, in particular the architectures of TMN systems. These reviews and associated theoretical work were able to set the scene for the experimental work. The project plan included a review and update of these studies after each experiment to add the benefits of the experience gained after the practical work and to keep up to date with the state of the art in advanced information processing technology and Telecommunications Management Network standards.

Experiment one. The first experiment was designed to solve the problem of correlated traffic causing buffer overflows and data loss, and hence reduced QoS. It dealt with the so-called "football game" problem, where several television stations are covering the same football match using the same set of cameras and using a variable bit rate coding mechanism to transport their pictures to the studio. Scene changes in the video traffic cause a burst of ATM cells. If the traffic for at least two of the television stations is routed over the same ATM link, correlated bursts are likely to cause buffer overflow and hence cell losses and potentially poor QoS for any traffic routed through that link. The management task is to detect correlated video traffic and to introduce a slight delay on one of the sources to remove the correlation before cell losses occur and QoS is degraded. This was seen as a relatively simple management problem to allow the experimental work to concentrate on developing the experimental infrastructure and the simulators—both being essential components of the next two experiments.

Because the project is investigating a particular network and service environment that is not operational today, simulation is a vital part of the experimental work. For the first experiment, NEMESYS required a simulation of an ATM network and a simulation of video and background traffic so that the management system could receive event reports from the simulated network; after making decisions, the system was able to initiate management actions in the simulator as if it were a real network. It is not necessary for the simulators to behave exactly as users and networks would in reality, but they should be sufficiently realistic for the purpose of demonstrating the functions and efficiency of the prototype management systems and the AIPs used to implement them.

There were two levels of simulation involved in the first experiment—One at the call level and the other at the ATM cell level. The call-level simulation was used for the majority of the experiment, and cell-level simulation was invoked when the detail of correlated video traffic streams was required.

Experiments two and three. In the second experiment, NEMESYS introduced a more comprehensive set of management functions that would perform more realistic and generalized traffic management compared to the relatively simple case in the first experiment. The management functions were broadly split into two areas: call acceptance management and virtual path management. Experiment two was intended as a preparation for experiment three.

For this second experiment the call-level simulation was unable to provide enough detail for the requirements of the management functions, and a cell-level simulation was too detailed, requiring a large processing overhead. For these reasons the network simulator was revised to simulate the transport of bursts of cells. A user simulator was added to model the behavior of users with regard to the services they used, the destinations they called, the frequencies of their calls, and the duration of their calls. A service simulator was inserted between the user and network simulators to model the traffic characteristics of the various services by generating bursts of ATM cells, the length and bandwidth of the bursts being dependent on the traffic type.

A set of candidate techniques, both conventional and AIP, were identified for implementing the management functions and one was chosen for each function. The intention was that alternative techniques would be used for the implementation of the same functions for the third and final experiment when comparisons could be made between the performance of the implementations against the evaluation criteria.

The third experiment uses the same management scenario as the second experiment, enhanced by a newly developed service management system. Figure 4 shows the functional components. The traffic manager and service manager are discussed in the next two sections, respectively. The management functions are described in some detail in the rest of this paper. During the design phase of this experiment we made extensive use of the open distributed processing approach that was developed as part of the standardization activities.[7] It helped us to structure the design space and to clearly separate the various design aspects.[8]

Traffic management

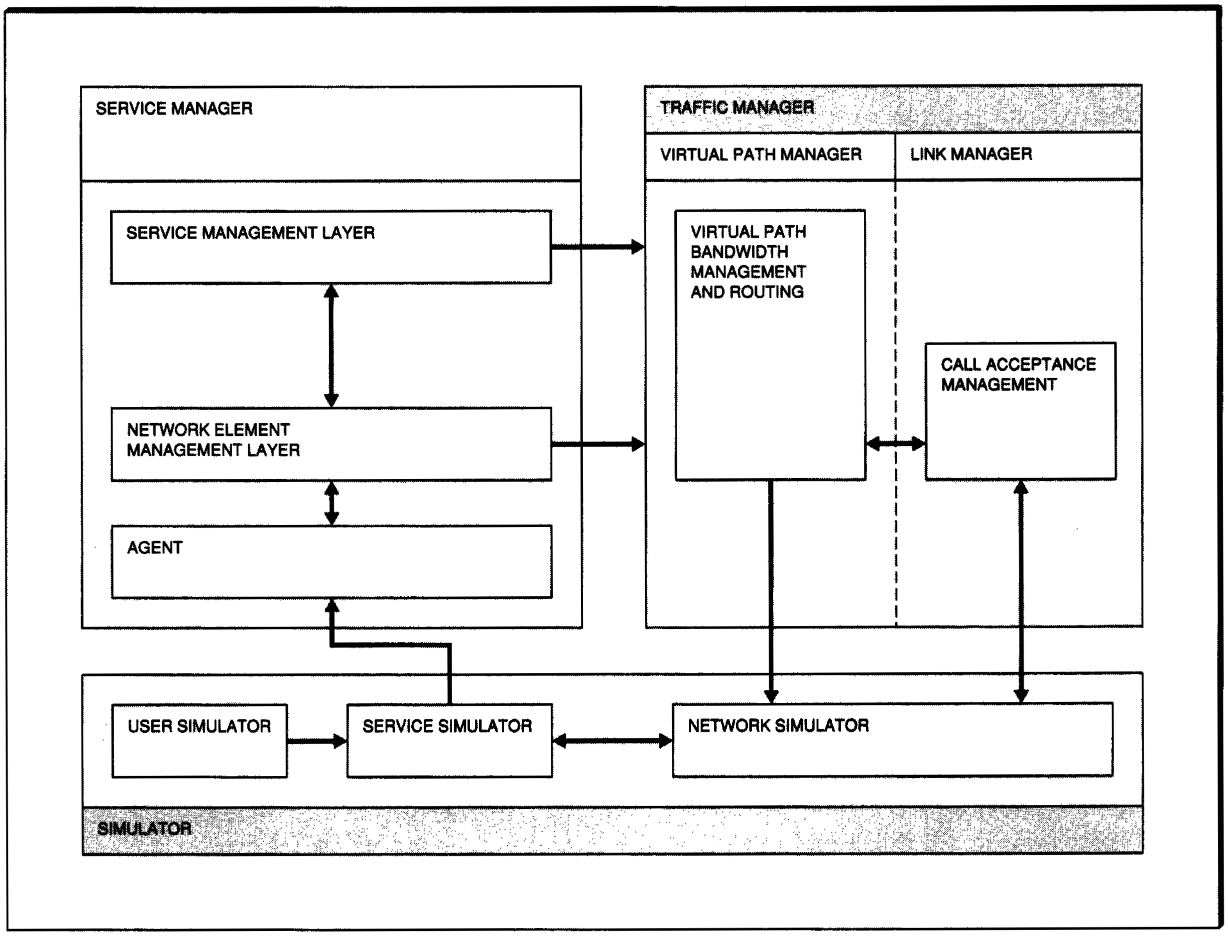
Experiment three of project NEMESYS investigates traffic management for ATM environments. Within this area we singled out two areas of research: call acceptance management (CAM) and virtual path management (VPM). Each area of management is described by a management control loop, whereby we mean a system that (1) becomes aware of a (bad) state, (2) takes a decision on which action to take among the great number of options, and (3) executes the chosen action.

Figure 5 illustrates the two management loops—the CAM loop and the VPM loop—and their interaction point, i.e., common managed objects. Solid lines are part of the loop; dashed lines provide input that may trigger the system to go through the loop.

The CAM loop. The purpose of the call acceptance management (CAM) loop is to allow as much traffic as possible onto the network without degrading the quality of service as seen by the customers. If calls are accepted according to their peak bandwidth (no statistical multiplexing) there will never be any losses of ATM cells, and the quality of service will be high but the network will always be lightly loaded. If calls are accepted according to their mean bandwidth (high statistical multiplexing), more calls can be accepted and the network will be more heavily loaded but there will be a large number of cell losses and hence a low QoS. The purpose of CAM is to strike a balance between these two extremes, i.e., to maximize the depth of statistical multiplexing while ensuring that quality of service does not fall to unacceptable levels. The mix of calls to achieve this balance is bounded by the so-called "feasible region." The purpose of the CAM loop is to determine this feasible region.

Under normal circumstances the service simulator sends call requests to the call acceptance function located in the network simulator. If the call, when added to the calls already established, falls within the feasible region, then the call is accepted; otherwise it is rejected. During a call the network simulator sends messages to the service

Figure 4 Components of experiment three



simulator for cells lost during the transmission. These cell loss reports are aggregated into reports that are sent to the service manager. The service manager then judges if the quality of service of the call was "good" or "bad" and forwards respective reports to the call acceptance manager. The judgment depends on the type of user, the service, etc. The CAM examines the QoS reports and, if necessary, adjusts the feasible region appropriately. The new feasible region is reported to the call acceptance function in the network simulator.

The VPM loop. The purpose of the virtual path management (VPM) loop is to maximize the proportion of successful connection attempts by optimizing the bandwidth reserved to virtual paths, the routes that they take, and the way connections are routed between them. A virtual path may traverse one or more links. A link is capable of transmitting a certain traffic load. If the load exceeds the maximum capacity, ATM cells are lost. However, this need not be the case with virtual paths, as the bandwidth of a virtual path is normally smaller than the bandwidth of its links. Indeed a link usually carries several virtual paths.

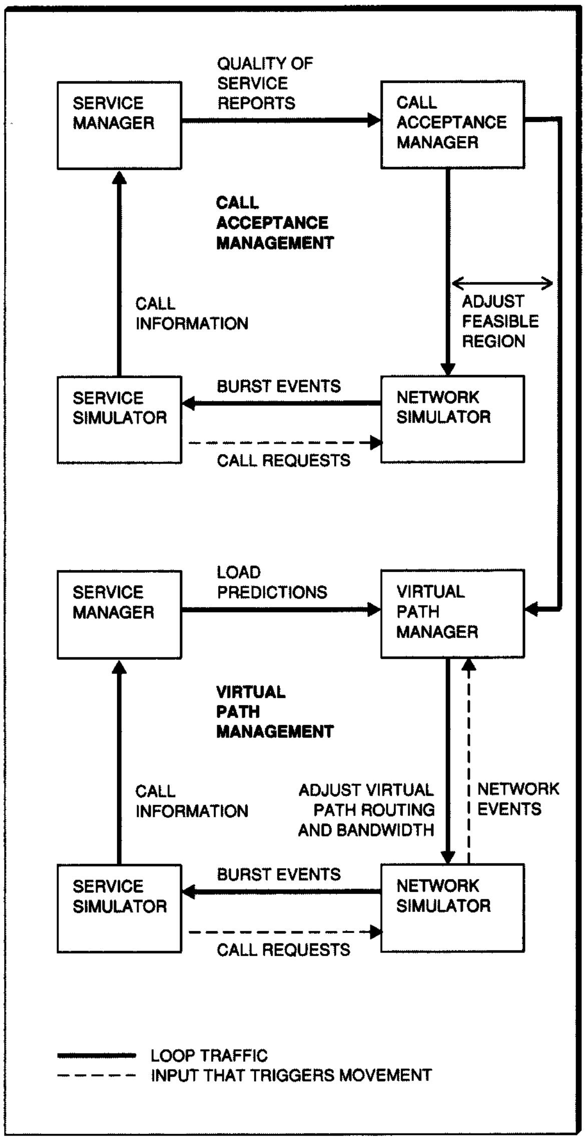
Routing table entries in the call acceptance function are used to route a connection between virtual paths on the way to its destination. The VPM loop manages both a network of virtual paths and the routing of connections between them. It can attempt to alter the reserved bandwidth of its virtual paths, and can change the relative priority of its routing table entries. Though it cannot create new virtual paths or routing table entries in the NEMESYS prototype, it can change the configuration of its virtual path network by disabling or enabling any of the preconfigured virtual paths and routing table entries.

The VPM loop works at the element, network, and service management levels, as defined in Reference 4. At the element level, it manages the bandwidth allocated to each virtual path, such that the amount of spare bandwidth is kept within bounds set by the network level. At the network level, it manages the routing of connections between virtual paths and the target amounts of spare bandwidth to be reserved for each virtual path, in order to conform to spare bandwidth targets set by the service level. The service level uses AIP techniques to set targets for the spare bandwidth to be reserved between the source and destination nodes of a virtual path.

The virtual path manager uses load predictions from the service manager, information about the feasible region from the call acceptance manager, and aggregated information about successful and unsuccessful connections from the network simulator in order to adjust the bandwidth allocation of virtual paths and routing of connections.

Advanced information processing for CAM. Let us first explain how the call acceptance manager manages the so-called feasible region of acceptable calls. First assume that only one type of connection can be accepted to the network, e.g., a telephone connection. Then the bandwidth of the link (the maximal bit rate) determines how many connections can be accepted. Today a voice connection is allocated a bandwidth of 64 000 bits per second (64 kbit/s). However, in future broadband networks based on ATM technology it may be that silence periods are not transmitted, and so the effective bandwidth of a voice connection is less than 64 kbit/s. Furthermore, the quality requirements may allow for loss of some ATM cells in every million of cells transmitted, and so the *effective bandwidth* may be smaller yet. These two reductions in bandwidth are a result of using another coding scheme, contrasting the current kind of pulse code modulation multiplexing where a pulse code modulation connection is allocated 64 kbit/s—no more, no less.

Figure 5 **The two management loops of traffic management**



Next assume that at least two different types of connections can be accepted on a network, e.g., a voice connection and a data transmission connection. For each kind of connection we can determine, either analytically, heuristically, or experimentally, what the effective bandwidth is. The feasible region where additional calls may be accepted then is a two-dimensional area that depends on a two-fold convolution of the statistical properties of each connection type; for N connection types the region is an N-fold convolution.

In order to decide whether a new call lies within the feasible region, we use and evaluate the following techniques: constraint programming, analytical/heuristic method, two-moments allocation scheme, neural network support, and peak and mean method.

Constraint programming—For this method it is assumed that the feasible region is bounded by a straight plane. To begin with, the feasible region is unbounded. As time progresses, reports are received telling of configurations that have shown bad quality of service. When a number of these reports have been received, the constraint programming tool is invoked. The implementation calculates the maximal feasible region that will exclude most of the points with bad quality of service. When a new batch of reports on bad QoS have been received, the constraint programming tool is invoked again. Currently, our implementation is able to decrease the feasible region only.

Analytic/heuristic method—For this method it is assumed that each type of connection is known by its mean and peak bit rate. We start with the (optimistic) assumption that calls may be accepted if a link has a spare capacity of at least the mean bit rate. Thus the border of the feasible region is initially based on the mean bit rates only. The (pessimistic) number of calls is obtained by assuming the peak rates for all connections. Whenever there is a bad QoS report during the operation of the system the borderline is moved in small, predetermined intervals toward the pessimistic border. Likewise, for a series of consecutive good QoS reports the borderline is moved back to the more optimistic side.

Two-moments allocation scheme—For this method we assume that a connection is known by its mean bit rate and its standard deviation, i.e., its two first moments. Theoretical work shows that if the bandwidth requirements of connections compared to the bandwidth of the link are small, then a linear combination of the mean and the standard deviation bounds the feasible region.[9] The method estimates the coefficients in the linear combination using buffer overflow probabilities.

Neural network support—For a neural network to work it must be trained. This is done by presenting both a connection configuration to the network and its quality of service. When the neural network has been exposed to many such configurations and QoS reports, it is able to provide decision support for the CAM.

Peak and mean method—We assume that the peak and the mean bit rate of a connection are reported to the call acceptance function. Based on this information the method determines an upper bound on the cell loss probability for the new configuration. If that cell loss probability is smaller than some given threshold value, then the connection is accepted.

The analytic/heuristic method is not an advanced information processing technique. However, it has been included in the experiment to serve as a reference against which all other methods can be judged.

For the peak and mean method our experiment shows that the method is singular for "nonbursty" traffic, that is, for traffic where the ratio of peak to mean bandwidth is equal to one. Therefore, a hybrid of the peak and mean method and some other method must be built to have a method allowing all kinds of traffic.

The results of the advanced information processing evaluation can be found in Reference 10.

For a description of advanced information processing techniques used for virtual path management in NEMESYS, see Reference 11.

Service management

Service management comprises activities to ensure the proper operation of services. Due to the variety of services provided in future Integrated Broadband Communications Networks, the management of these services will be a complex task involving a number of organizational entities. The notion of quality of service is of fundamental importance here.

This section concentrates on architectural structures and management standards that may become relevant for public as well as customer premises networks. The evaluation of advanced information processing techniques is of minor concern.

Service model. The service model assumes a layering of services. A service may itself make use of underlying services. The model is analogous to the notion of service in the OSI Reference Model.[12] For example, a multimedia information service needs database services as well as communication services. Each of the services has its own QoS parameters. Service providers may also be service users, and thus depend on lower layer QoS parameters, whereby a QoS parameter of the layer below may directly influence the QoS on the higher layer. For example, an increased cell loss rate on an ATM link will have a negative effect on the transmission delay due to possible retransmissions.

Service quality generally depends on a number of different QoS parameters. Definition of acceptable QoS parameter ranges is part of the service contract between the service user and service provider, either explicitly or implicitly. It is then the responsibility of the service provider to make sure that specified QoS values are met during the provision of the service. The contract would define how potential QoS violations are handled. Additionally, a service contract would contain other information such as service types, access points, duration, cost, user obligations, provider obligations, and more.

The following is an overview of the NEMESYS service management assumptions which, although based on CCITT documents and common telecommunication knowledge, represent the authors' perspective on future issues of IBCN management. For a complete detailed specification of the NEMESYS experiment, see Reference 11.

Service users establish *service associations* with service providers. Acceptable QoS values, ranges, or thresholds are specified for the respective service parameters. The service provider builds its service on top of underlying network services offered via network connections. The provision of services is managed by a service manager, which is a distributed application.

Service manager functions. We assume that the service manager performs the functions of service user administration, QoS monitoring and control, event logging and statistics, profile maintenance, and load evaluation and prediction.

Service administration stores and updates information about services, their providers, and QoS parameter values that can be guaranteed to users. Trading aspects, such as which service provider offers the cheapest service given the QoS requirements, are not addressed. User administration functions cover all aspects of information about end users subscribed to services, e.g., names, addresses, which service providers are under contract.

QoS monitoring and control ensures that users get the QoS they subscribed to (and pay for). A possible countermeasure in case of bad QoS levels may be, for example, to tell the network service provider that a particular network service QoS parameter has gone bad, causing a poor service QoS value.

Event logging and statistics are important management functions, which help to analyze the usage patterns and systems behavior over some period of time. This is needed to plan the future allocation of resources. Various forms of measurement data aggregation are computed to obtain insights into the system performance.

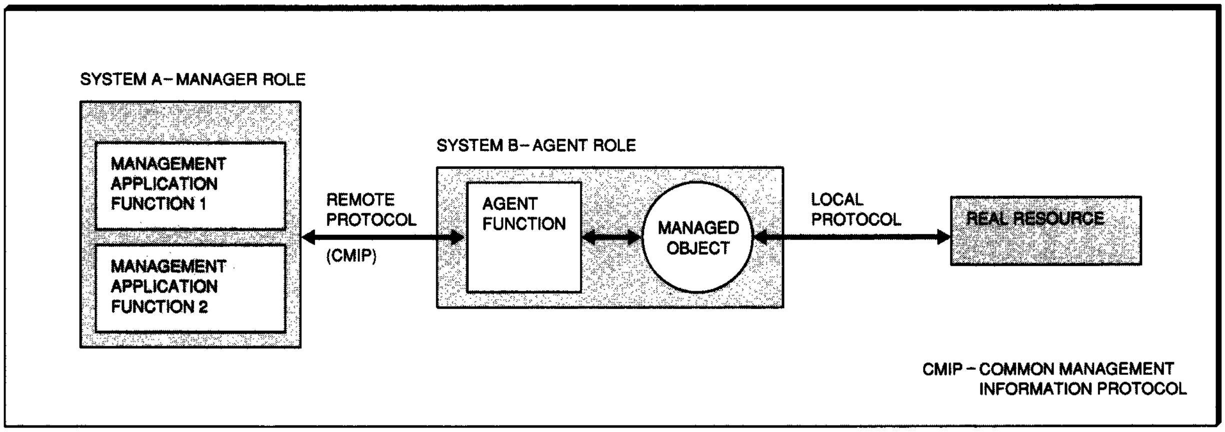
Profiles store information on the behavior of entities such as users, services, and access points. They also contain threshold values for QoS parameters.

Load evaluation covers (short- and long-term) planning aspects to enable good QoS in the future. The service provider needs to estimate the load that users will generate at some time in the future, in order to plan its own resource requirements and to use lower layer services efficiently. For example, it might be necessary to ask a network service provider for more transmission capacity well before it is actually needed. The latter is a *proactive* management approach as opposed to a mere *reactive* approach.

Service types. In order to reduce the complexity of the experiment, we have made simplifying assumptions for the prototype as to what services are considered, i.e., we concentrate on communication-oriented services only, excluding other potential service components such as database services. The complete detailed definition is contained in Reference 11.

NEMESYS assumes seven types of services that users can request:

Figure 6 Standard conformant management communication



- Pulse code modulation telephone (fixed bandwidth coding scheme)
- Time assignment speech interpolation telephone (variable bandwidth coding scheme)
- Interactive data transfer
- Bulk file transfer
- Video broadcast
- Video conference
- Multimedia document transfer

Each of the user services is built upon a given combination of network services, of which there are four types: constant bitrate sound, variable bitrate sound, data, and full-motion video.

Two examples for the kind of mappings of user services to network services are video conference and multimedia document transfer. Video conference uses two *constant bitrate sound* and two *picture* network connections, i.e., one (*sound, picture*) pair per direction. *Multimedia document transfer* uses one *data* network connection to transmit the request from the user to the service provider (e.g., a multimedia database service) and one *constant bitrate sound*, one *data*, and one *picture* network connection to receive the document back.

Quality-of-service parameters. For a service association we use five quality-of-service parameters: establishment delay, signal delay, signal delay variation, disconnect delay, and signal quality.

The quality-of-service parameters of a network connection are cell loss rate, establishment delay, signal delay, signal delay variation, disconnect delay, and signal quality.

The network QoS parameters are used to compute the user service QoS values. For example, establishment delay is defined to be the maximum of the establishment delays of the corresponding individual network connections. For further details, see Reference 11.

Standard conformant service management implementation. For the implementation of the experimental service manager we investigate advanced information processing techniques such as the use of object-oriented design and implementation techniques, as well as the integration of an object-oriented database system. Equally important, however, is the exploitation and experimentation with standard conformant management solutions.

As the Telecommunications Management Network standard[4] emerges, it becomes clear that management functions for the IBCN will make use of OSI management models and protocols. Thus, we designed and implemented the service management system presented in the previous section, as close to the standards as possible (within the limitations of a prototype).

Network management standards. All standardization groups have adopted an object-oriented approach. Real resources—physical ones such as modems, users, and service providers, and logical ones such as connections, services, and profiles—are modeled by so-called *managed objects*.[5] A managed object provides an abstract view of the resource. It is the responsibility of managed objects to handle possible interactions to the real resource they represent (see Figure 6). This interaction has been deliberately left outside standardization as it is a matter of local implementation. Any method can be used, such as shared memory, local interprocess communication mechanisms, proprietary management protocols, or the Simple Network Management Protocol (SNMP).[13]

Management application functions manipulate managed objects (create, delete, get, set, activate actions, receive events) in order to perform management tasks. If management application functions and managed objects reside together, interaction is implementation-dependent. For the remote case (see Figure 6), standardization has defined a common management information service (CMIS) and protocol (CMIP).[14,15]

During interaction, systems may take different roles. *Managers* perform management application functions. To do so, they manipulate managed objects. As managed objects do not support management communication functions, managers access *agents* that forward requests to the corresponding managed object(s). It is possible to manipulate many managed objects by one request only, e.g., to read the status of several devices. Two systems may change their roles for different interactions. System A in Figure 6, taking the manager role for accounting management, for example, may take the agent role for performance management later.

Managed objects are organized in three different ways. They are defined in an inheritance tree, specifying which class of managed object inherits properties from another class (making use of object-oriented techniques). A containment tree defines the names of instantiated managed objects (representing resources) in some running management system. The conceptual repository of all management information, i.e., all instantiated managed objects, is known as the Management Information Base.[5] Finally, the registration tree assigns unique names to managed object classes so that they can be understood by management applications written from independent organizations.

The OSI management model strongly encourages the implementation of event-driven management applications. Managed objects are able to send events concerning the operation of the real resource they represent, e.g., to report a status change or performance degradation.

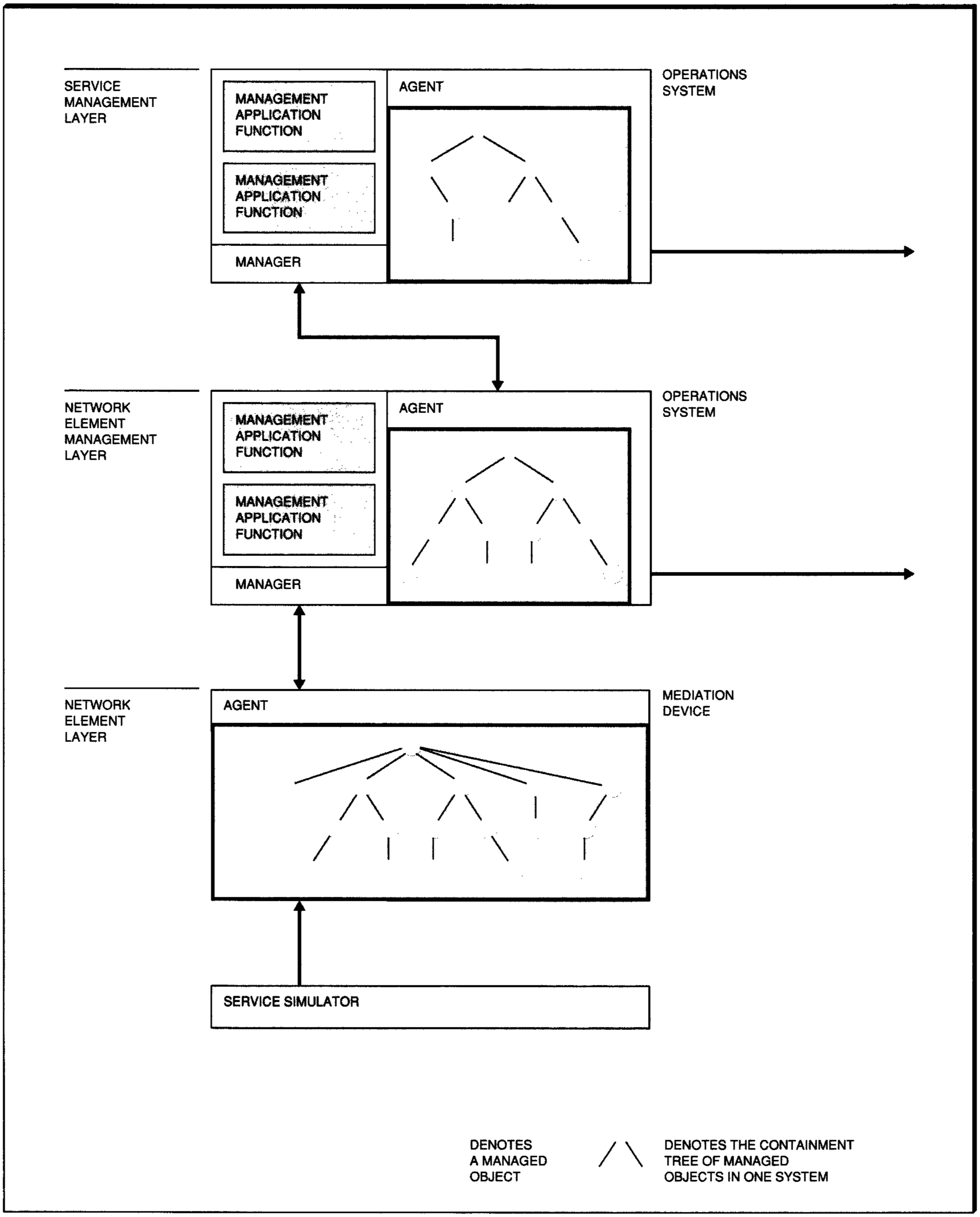
Service manager design. Based on the envisaged service management function as described in the previous sections and based on our experience from the first two experimental studies of NEMESYS, we designed the following service management system (see Figure 7).[16] As a real IBCN environment is not yet operational, user functions (initiating calls), service functions (setting up network connections, generating the load of calls) and network functions (routing, call acceptance, data transport) are simulated.

On top of the service simulator we implemented an agent offering an interface conforming to management standards. In the terms of Reference 4 this would be called a *mediation device.* It offers access to managed objects[17] such as service associations that represent the call of an end user and network connections that represent a network connection offered by some network provider. Service associations and network connections are created and deleted depending on the service simulators activity. Implementing a private interface, we selected a proxy management approach instead of modifying the service simulator to provide this management interface.

On top of the service simulator agent we implemented an intermediate system (an operations system with different management application functions) that manages the agent, e.g., it supervises service associations or network connections, updates threshold values in the profiles, and alerts the network operator if the quality of network connections is not acceptable when measured against a contract.

The operations system also aggregates information about service associations and network connections, such as the number of service associations established and refused during some settable time interval, and the mean value and

Figure 7 Service management architecture



standard deviation of service performance parameters as service association setup delays.

This information is stored in appropriate managed objects emitting events about aggregated information periodically. In our prototype we implemented a third system (another operations system with different management application functions) on top of the intermediate one using its events to perform the following tasks:

- Administrate end users implementing managed objects representing end users and addresses.
- Administrate services implementing managed objects representing services, service provider, etc. One typical management application function is the calculation of performance parameters that can be guaranteed to new users. These performance parameters are based on measured values sent by the intermediate management application.
- Predict the number of network connections that will be required in the near future. This is done by the service manager to help the traffic manager, as it has better knowledge about user behavior.

Implementation decisions. The management application function (Figure 7) encapsulates the intelligence needed to map from the lower level information model (e.g., service associations) to its own (e.g., service access points).[16] Though parts of a management application function may be implemented in the managed objects, we separated management application functions and managed objects in our local implementation. We were thus able to reduce development time by parallel implementation.

Our work is based on a well-known management infrastructure named OSIMIS,[18] which has attracted widespread attention from many research institutes as well as commercial companies. It provides common management information protocol communication based on ISODE,[19] the public domain OSI stack, and a generic agent infrastructure. This work was extended significantly for the needs of the NEMESYS prototype. It now provides a generic management infrastructure supporting a convenient programming of management application functions. The interfaces from management application functions to managed objects residing in the same system, as well as those residing in other systems—and thus accessible via common management information service only—are modeled. Both interfaces are object oriented and very similar.

Evaluation. We found the structure recommended in the TMN architecture appropriate for our prototype. The functional layering divides the overall management task into smaller pieces, thus reducing the complexity during design and implementation. We found the use of such a hierarchical layered decomposition important in order to distribute the work load among partners. This was facilitated by well-defined interfaces between layers and functional components.

The use of the ISO/OSI management standards influenced the design of management application functions. The event-driven approach suggested by those standards compared to the polling approach used in the NEMESYS traffic manager turns out to be more efficient. Management applications consume resources only when relevant events occur.

However, we found a good infrastructure support extremely important in order to construct an ISO/OSI-based management system, e.g., to implement management algorithms, to handle managed objects, and to run and control the system.

Conclusions

Management solutions for the future IBCN and its multiple services pose challenging research problems. In this paper we have presented some important aspects of traffic and quality-of-service management. The conclusions cover three domains: advanced information processing techniques, architecture, and management standards.

For advanced information processing (AIP), the use of object orientation for design and implementation is a very convincing conclusion. It has been applied broadly throughout the prototype and the results are very positive: designers and developers are enthusiastic, their productivity is improved, and the resulting programs are efficient and easy to maintain. The fact that more managed objects are being defined by the standards organizations make their reuse in the development phase quite coherent.

The distribution of management functions at several functional layers, as recommended by CCITT

M.3010,[4] for instance, proved to be the right direction but not easy to implement in practice. Many choices are necessarily left to the designer and they are not straightforward, e.g., how to assign managed objects to different functional layers. A second set of problems was explored in the domain of cooperation between several independent network and service managers. This proved to be complex, and we progressed on the definition of what really needs to be exchanged between them. The confrontation of the architecture—which is abstract—to detailed real-life cases is essential to provide an in-depth common understanding of its conceptual validity and eventually to make it more precise. The use of the common management information service and protocol itself raised no problem.

The development of the asynchronous transfer mode (ATM) network and services simulators gave us an in-depth understanding of the ATM characteristics. More real-life data would be needed to validate the traffic patterns and the required quality parameters of the various services we simulated.

Until the completion of project NEMESYS, we will continue to enhance our prototype. Some results of this project have already been carried over to other projects, where the simulators will be replaced by a real high-speed networking environment.

Acknowledgments

This work was supported by the Commission of the European Communities under the RACE program. The authors wish to thank the partners of the RACE NEMESYS project who contributed to these results: Dowty Communications Ltd., Fischer & Lorenz AS, GPT LIMITED, Institute for Computer Science of Crete, KTAS Copenhagen Telephone Company, University College London, GSI-Erli SA, and IBM Europe.

Cited references

1. P. J. Kühn, "From ISDN to IBCN (Integrated Broadband Communications Network)," *Proceedings of the XI IFIP Congress '89, Information Processing 89*, G. X. Ritter, Editor, Elsevier Science Publishers B.V., IFIP (1989), pp. 479–486.
2. *Draft Recommendation I.150: B-ISDN ATM Functional Characteristics*, Study Group XVIII, International Telecommunication Union—CCITT, Geneva (1990).
3. *Recommendation E.800: Terms and Definitions Related to the Quality of Telecommunication Services*, CCITT Blue Book, International Telecommunication Union—CCITT, Geneva (1988).
4. *Draft Recommendation M.3010: Principles for a Telecommunications Management Network*, Study Group IV—Report R 2, International Telecommunication Union—CCITT, Geneva (1991).
5. *Information Technology—Open Systems Interconnection—Systems Management Overview*, ISO/IEC International Standard 10040, International Organization for Standardization, Geneva (1991).
6. NEMESYS consortium, *Specification of the NIKESH Experimental System*, NEMESYS, RACE Project No. 1005, Document No. 05/GPT/BNM/DS/X/009/b1, D. Griffin, Editor, A. Galis, contact, DOWTY Communications Ltd., Caxton Way, Watford, Herts WD1 8XH, UK (1990).
7. *Draft Documents of the Basic Reference Model of Open Distributed Processing*, ISO/IEC JTC1/SC21/WG7 (the standards will be called IS 10746 and X.900 by ISO and CCITT, respectively), International Organization for Standardization, Geneva (1991).
8. K. Geihs and A. Mann, *ODP Viewpoints of IBCN Service Management*, Technical Report No. 43.9104, IBM (European Networking Center), A. Mann, contact (1991).
9. C. Courcoubetis, G. Fousakas, V. Friesen, and S. Sartzetakis, "Real-Time Issues in Call Acceptance Management for ATM Networks," *Proceedings of the Fifth RACE TMN Conference*, London, Nov. 20–22, A. Mamdani, contact, Queen Mary & Westfield College, UK (1991), pp. III.1/3.1–III.1/3.12.
10. A. Gach, C. Miarlaret, and P. E. Allard, "An Experimental Evaluation of Call Acceptance Management Algorithms in ATM Based Networks," *Proceedings of the Canadian Conference on Electrical and Computer Engineering (CCECE-92)*, Toronto, Canada (Sept. 13–16, 1992).
11. NEMESYS consortium, *Emmanuel—Case Study, Specification and Runs, Initial Design*, NEMESYS, RACE Project No. 1005, Document No. 05/KT/AS/PI/C/020/b2, A. Galis, contact, DOWTY Communications Ltd., Caxton Way, Watford, Herts WD1 8XH, UK (1992).
12. *Information Technology—Open Systems Interconnection—Basic Reference Model*, ISO/IEC International Standard 7498, International Organization for Standardization, Geneva (1984).
13. J. Case, M. Fedor, M. Schoffstall, and J. Davin, *A Simple Network Management Protocol (SNMP)*, Request for Comment (RFC) 1157, Network Information Center, SRI International, Menlo Park, CA (1990).
14. *Information Technology—Open Systems Interconnection—Common Management Information Service Definition*, ISO/IEC International Standard 9595, Version 2, International Organization for Standardization, Geneva (1991).
15. *Information Technology—Open Systems Interconnection—Common Management Information Protocol Specification*, ISO/IEC International Standard 9596, Version 2, International Organization for Standardization, Geneva (1991).
16. U. Harksen, A. Mann, and G. Pavlou, "Experience of Modelling and Implementing a Quality of Service Management System," in *The Management of Telecommunications Networks*, R. Smith, A. Mamdani, and J. Calla-

ghan, Editors, Ellis Horwood Ltd., Chichester, West Sussex, UK (1992), pp. 109–120.
17. *Information Technology—Open Systems Interconnection—Structure of Management Information—Part 1: Management Information Model*, ISO/IEC International Standard 10165-1, International Organization for Standardization, Geneva (1992).
18. G. Knight, G. Pavlou, and S. Walton, "Experience of Implementing Network Management Facilities," *Proceedings of the 2nd Symposium on Integrated Network Management*, II, I. Krishnan and W. Zimmer, Editors, Elsevier Science Publishers B.V., IFIP (1991) pp. 259–270.
19. M. T. Rose, *The Open Book*, Prentice-Hall, Inc., Englewood Cliffs, NJ (1990).

Accepted for publication August 3, 1992.

Kurt Geihs *University of Frankfurt, Computer Science Department, P.O. Box 11 19 32, D-6000 Frankfurt, Germany (electronic mail: geihs@informatik.uni-frankfurt.de).* Dr. Geihs has been a professor of computer science at the University of Frankfurt since April 1992. His research and teaching focuses on distributed systems and operating systems. Before joining the university he worked for IBM at the IBM European Networking Center in Heidelberg, Germany. His research areas were network operating systems, open distributed processing, and system management. In 1988–89 he was on assignment to the IBM Thomas J. Watson Research Center in Hawthorne, New York, developing software for high-speed network attachments. Dr. Geihs received a Diplom-Informatiker degree from the Technical University Darmstadt, Germany, an M.S. in computer science from the University of California, Los Angeles, California, and a Ph.D. from the Technical University of Aachen, Germany.

Philippe Francois *IBM France, 06610 La Gaude, France (electronic mail: francoip@lgemvxa.vnet.ibm.com).* Mr. Francois is currently a networking consultant at the IBM Net-Review International consulting service. He received the M.S. in applied mathematics from Paris University. His first professional experience was as a member of the French National Research Center (CNRS) where he worked on numerical analysis algorithms. He joined IBM at the La Gaude development laboratory in 1965. From 1965 to 1975 he worked as designer and developer of the IBM PABX software, and in 1972 he became manager of the IBM PABX software development. From 1976 to 1984 he was an architect and then became manager of the La Gaude SNA architecture department, working mainly on X.25 and OSI support by SNA. He was assigned to the 3745 Communications Controller planning and architecture department in 1985, and in 1991 was responsible for the IBM participation in two European-sponsored RACE projects dedicated to communications management in broadband ATM systems.

David Griffin *Network Systems Group, GPT LIMITED, New Century Park, P.O. Box 53, Coventry CV3 1HJ, United Kingdom (electronic mail: briad_brandist@eurokom.ie).* Mr. Griffin joined GEC Telecommunications Ltd., a division of General Electric Company plc, in 1984 as a sponsored student. In 1988 he received a B.Sc. in electronic, computer, and systems engineering from Loughborough University of Technology in Leicestershire, United Kingdom. He joined GPT (GEC Plessey Telecommunications) in 1988, and in 1989 he was employed as a telecommunications systems engineer working on the Commission of the European Communities-sponsored RACE program in the NEMESYS project where he contributed to the telecommunications systems modeling work. In 1991 he was elected chairman of the project's technical committee. In addition he has represented the project on the RACE working group on TMN architecture and interfaces.

Christian Kaas-Petersen *KTAS (Copenhagen Telephone Company), Research and Development, Norregade 21, 1199 Copenhagen K, Denmark (electronic mail: ckp@ktas.dk).* Dr. Kaas-Petersen has worked for KTAS since 1989, where he has been involved in RACE projects funded by the European Economic Community. These RACE projects have been on the topics of quality of service and network management. Before joining KTAS, Dr. Kaas-Petersen worked as a post-doctoral research Fellow at the University of Leeds, England, on computational methods for bifurcation analysis of nonlinear dynamical systems.

Andreas Mann *IBM European Networking Center, Tiergartenstrasse 8, P.O. Box 10 30 68, D-6900 Heidelberg, Germany (electronic mail: mann@enc.ibm-deutschland.dbp.de).* Dr. Mann is currently a research staff member at the IBM European Networking Center. Since joining IBM two years ago he has worked in the area of systems management, focusing on future high-speed networks' integrating of different services. His main research areas include standard conformant management systems and quality-of-service management, and he often gives lectures about these topics at the Technical University of Aachen, Germany. Dr. Mann finished his diploma thesis in 1985 at the University of Karlsruhe, Germany, working on data security. From 1987 to 1989 he worked at the Technical University of Aachen, Germany, on mobile communications and performance analysis; in 1989 he received his Ph.D.

Reprint Order No. G321-5494.