INTERNATIONAL JOURNAL OF DIGITAL AND ANALOG COMMUNICATION SYSTEMS, VOL. 3, 229–238 (1990)

AN ARCHITECTURE FOR INTEGRATED NETWORKS THAT GUARANTEES QUALITY OF SERVICE

AUREL A. LAZAR, ADAM TEMPLE AND RAFAEL GIDRON

Department of Electrical Engineering and Center for Telecommunications Research, Columbia University, 500 W 120th St., New York, NY 10027-6699, U.S.A.

SUMMARY

A generic architecture for integrated networks that guarantees quality of service is described. The network has a mesh topology and a switching architecture structured according to the concept of asynchronous time sharing. This concept is based on a multiclass network model and asynchronous algorithms for allocating network resources. There are three traffic classes for transporting user information and a fourth class for network management and control. Resource allocation is resolved through time-sharing and space-partitioning algorithms.

KEY WORDS Quality of service Switching architectures Scheduling and buffer management

1. INTRODUCTION

The design of most telecommunication networks has, from the user's point of view, a major flaw: it cannot efficiently guarantee *quality of service*. To the best of our knowledge, the concept of quality of service does not *explicitly* appear in the design specifications of integrated networks, and thus their *performance* appears to be an afterthought.[1] For example, traffic classes with their associated attributes have not been explicitly taken into account in existing exploratory designs. Adaptivity of network control parameters to the traffic load and profile has received very little attention.

In 1985, we started a research programme with the goal of understanding the behaviour, design and implementation issues of integrated networks from the *performance* point of view. From the start, we took into account the impact that traffic control architecture (TCA) requirements might have on the hardware design. Based on our experience during this programme,[2–6] a set of network design principles emerged. These principles are described in this paper.

The following TCA requirements have been considered. First, guaranteed quality of service.[2] We envision that the integrated network will offer a guaranteed quality of service as negotiated at call set-up. Sessions that do not require a call set-up will not receive a specified quality of service. Secondly, adaptive user-defined networks. The network might support large users who require virtual private networks. Whereas today the user-defined networks are largely set up by the users themselves, we envision that the traffic control architecture of these networks will automatically set up virtual networks. Thirdly, prediction capabilities. For example, a mobile user who logs in at one point into the network might need to be guaranteed a quality of service while he/she is spatially moving in time. Predicting the path of the user can lead to the appropriate resource allocation and control policy. To achieve this goal, new learning algorithms will be needed.

In order to design and implement integrated networks that guarantee a quality of service as negotiated at call set up, a performance-orientated concept called *asynchronous time sharing* (ATS) is formally proposed here as a design principle for integrated networks.[7] This concept is based on a multiclass network model and asynchronous algorithms for allocating network resources.

The network model has four classes of traffic. Class C supports information transfer for network management and control. Classes I, II and III support user traffic. Class attributes are defined by a set of quality of service parameters. The resources considered here are switching bandwidth, communication bandwidth and buffer space. Access to switching and communication resources is resolved through a scheduling algorithm based upon time sharing. At each switching node or communication link, the four traffic classes share these resources sequentially in time. Buffer management is achieved via space partitioning. The efficiency of the network, operating under the constraints imposed by the quality of service requirements, is measured by the average throughput and the probability of blocking at the call level.

A generic structure of an integrated switching node that implements the ATS concept is presented. The basic architecture consists of a switch fabric interconnecting groups of input and output buffers. The input buffers provide queueing space for all

1047–9627/90/020229–10$05.00

Received 14 December 1989
Revised 25 April 1990

four traffic classes at each access point to the switch fabric. The output buffers perform a similar function at the access point to each communication link. The switch fabric must be non-blocking under the loading or operating conditions of the network.

This generic switching architecture is novel in two ways. First, the concept of quality of service for multiple traffic classes explicitly appears in the design specifications at both the edge and the core of the network. Therefore, one of our fundamental requirements is that the *core* of the network makes a distinction between traffic classes. This is necessary to provide guaranteed quality of service efficiently. Note that this is not a requirement of ATM-based integrated networks. Secondly, network management primitives are incorporated into the switching architecture in hardware (data link layer). For example, traffic monitoring and associated statistics at a switching access point are evaluated in hardware as part of an intelligent resource allocation system.

This paper is organized as follows. Fundamental issues arising in congestion control of integrated networks are presented in Section 2. In Section 3 an integrated network model that is capable of overcoming the limitations of the existing architectures is described. The multiclass network model is presented in subsection 3.1. Network and user performance parameters are described in subsection 3.2. and a general resource allocation concept in subsection 3.3. In Section 4 the basic switching architecture is introduced. The system model of the switching architecture is detailed in subsection 4.1. The implementation of the asynchronous time sharing principle and its operation are presented in subsection 4.2.

2. MOTIVATION: TRANSIENT CONGESTION PHENOMENA IN INTEGRATED NETWORKS

Packet-switching networks allow a flexible and bandwidth-efficient implementation of services. This flexibility and efficiency, however, has a price in terms of implementation complexity of the traffic control architecture. For these networks, the problem of guaranteeing quality of service appears to be much more difficult to solve than in circuit-switched networks. In what follows, some transient congestion phenomena that might arise in packet-switching networks will be discussed. Three fundamental issues that arise in the multiplexing and switching of integrated traffic are examined (see Figure 1).

First, consider the simple multiplexing problem depicted in Figure 1(a). Two traffic flows (voice and data) access a common buffer of an integrated node. We will assume that a 250 ms burst of data packet arrives at very high speed to the input of

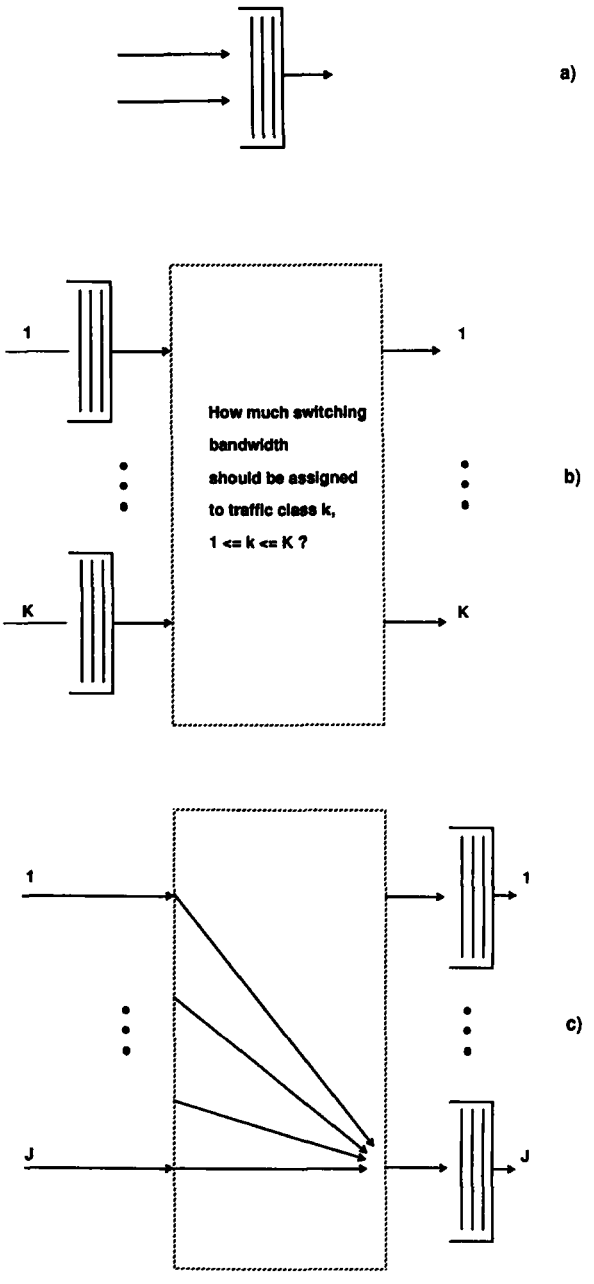


Figure 1. Fundamental congestion issues arising in multiplexing and switching of integrated traffic

the buffer. We will also assume that, owing to the burst arrival, the buffer will be in overload during this time period. Thus, the voice packets will not be able to access the buffer and will be blocked.

What will be the resulting quality of service for the two traffic flows? Since data packets are transmitted at a very high rate and the buffer has a finite size, some of its packets will be blocked. An end-to-end retransmission protocol, however, will guarantee that no packets will be lost. Thus, the quality of service degradation for the data packet flow is manifested through a longer average time delay. This might either be unnoticeable or not relevant (from the users point of view). But what

about the voice packet flow? Since the buffer will be full for a transitory period of 250 ms, all arriving packets will be blocked. Given that real-time protocols do not allow for retransmissions, all voice packets during the 250 ms period will be lost. This might lead to a serious degradation in the quality of the voice call.

The second example is depicted in Figure 1(b). Consider a switching node in an integrated network with K generic traffic classes that are characterized by different quality of service requirements. For example, real-time traffic such as voice or video has stringent time delay requirements. Data traffic, on the other hand, is less sensitive to time delay. The switching bandwidth is shared among all traffic classes. The fundamental problem is to provide, for a given traffic load and profile, the appropriate switching bandwidth allocation among the classes that guarantees the quality of service for each class. For instance, if motion video traffic is not given the appropriate switching bandwidth, its quality of service is degraded since packets cannot be served in a timely manner. Allocation for bursty types of traffic that is based on the peak bandwidth requirement (as in circuit-switched networks) results in an inefficient usage of network resources. This becomes critical during periods of congestion.

A third example is shown in Figure 1(c). In this case, multiple input access points of a switching node transmit packets to a single output port. For simplicity, we assume here that the same type of traffic (e.g. video) accesses output port number J. It is easy to see that when transient congestion occurs, the output buffer associated with port J becomes full. Thus, this scenario will also cause lost packets, with a corresponding degradation in the quality of service.

Note that typical implementations, such as interconnection networks or distributed switches (such as LANs), do not address the fundamental output port congestion issue that arises in an integrated environment. They are basically limited to solving the in-switch routeing problem. In the case of interconnection networks, each input access point typically sends an equal number of packets into the switch during a given amount of time. However, the input access points do not carry the same traffic load destined for the different output ports. Note also that a priority mechanism on traffic flows divided into traffic classes *cannot* resolve the output port fairness issue.

The basic solution proposed in this paper for the fundamental multiplexing problem described in the first example is to store the different traffic types into logically separate buffers. This requires, however, that the core of the network makes a distinction between traffic classes. In the second example, providing for flexible scheduling of resources to different traffic classes is the key to efficiently guaranteeing quality of service to users. Contention between different traffic classes is resolved using a time-sharing algorithm. The time of occupancy (service time) of a resource such as a switch fabric or communication link is dynamically controlled. This requires programmability of switching and multiplexing resources at each node in the network. The solution to the third problem is to control the allocation of the number of packets that each access point is allowed to send into the switch (per unit of time for each traffic class and output port). In the following sections, a general network model as well as resource-sharing mechanisms are proposed that can guarantee quality of service for different traffic flows on a network-wide basis.

3. THE INTEGRATED NETWORK MODEL

The integrated network considered in this paper has a mesh topology and transports services such as video, voice, data, graphics and facisimile. The core of the network does not make a distinction between these services. Instead, it recognizes a set of well-defined traffic classes. The user, prior to negotiating the quality of service, maps his application into one (or more than one) of these classes.

Four classes of packets are defined. The class is an abstract concept that is specified through delay and loss characteristics. One class of packets supports network management and control traffic. The other three classes transport user traffic. Characterization of these traffic classes is given in subsection 3.1. below.

A set of performance measures define the attributes of the three user classes. The *average throughput* and the *probability of blocking* for each of these is used as a measure of efficiency of the network. These measures are presented in subsection 3.2. The resource allocation algorithms proposed here require a time sharing of the switching and communication bandwidth among the four traffic classes. They also require buffer space partitioning among the classes. These fundamental mechanisms for ensuring quality of service are described in subsection 3.3.

3.1. *The multiclass network model*

As mentioned above, the multiclass network model considered in this paper supports four classes of traffic. Three of the traffic classes, Classes I, II and III, transport user traffic and are defined by a set of performance constraints. The fourth class, Class C, transports traffic of the network management system. While no formal performance criteria are associated with Class C traffic, it is assumed here that packets belonging to this class will not encounter congestion in the network. This can be achieved by proper allocation of network resources.

That is, by reserving sufficient switching and communication bandwidth for Class C, the network can ensure that this traffic will encounter only negligible queueing delays.

Class I traffic is characterized by 0 per cent *contention packet loss* and an end-to-end time delay distribution with a narrow support. The maximum end-to-end delay between the source and destination stations is denoted by S^{I} (see Figure 2(a)). Class II traffic is characterized by ϵ per cent contention packet loss and an upper bound, η, on the average number of consecutively lost packets. It is also characterized by an end-to-end time delay distribution with a larger support than Class I. The maximum end-to-end time delay is S^{II} (see Figure 2(b)). Here, ϵ and η are arbitrarily small numbers and $S^{I} \leqslant S^{II}$. *Contention packet loss* represents packets that are clipped or blocked. Clipping refers to packets that, because of network congestion, had an end-to-end delay greater than the maximum limit (S^{I} or S^{II}).[8] Blocking refers to packets that were discarded within the network by a buffer management system due to buffer overflow. For Class I and II traffic, there is no retransmission policy for lost packets. Class III traffic is characterized by 0 per cent *end-to-end packet loss* that is achieved with an end-to-end retransmission policy for error correction. If requested, it is also characterized by a *minimum* average user throughput Γ and a *maximum* average user time delay T (see Figure 2(c)). Thus, objectively quantifiable measures have been associated with the quality of service requirements, and a user can map a particular application to the appropriate traffic classes.

The abstraction of the concept of traffic classes is the result of our experience over the years with the implementation of services in an integrated local area network (ILAN) environment.[5, 6, 9] Class I guarantees a service comparable to that in circuit-switched networks. Class II guarantees a service with limits both on the packet loss and on the average of consecutively lost packets that is caused by finite buffer sizes and the time-delay constraint. It is of interest to video and voice sources where some packet loss is acceptable.[10, 11] Class III supports two types of service. The first type, where no specific quality or service is requested, typically supports datagram applications such as data and source code file transfers. The second type, where a specific quality of service is requested, guarantees a minimum average throughput and a maximum average time delay. Applications requesting such a quality of service can be supported by virtual circuits. Examples are image and graphic file transfers.

The existence of the four traffic classes leads to the existence of four *virtual networks*. Each of these networks supports traffic associated with a particular traffic class (see Figure 3). Thus, on a logical level four networks transport information in parallel. The events in these networks (such as packet service times) are strongly correlated because they use the same switching and communication facilities. The relative division of the switching and communication bandwidth, and the buffer space between the four virtual networks, is a key issue to be resolved by the TCA of the integrated network. It is determined

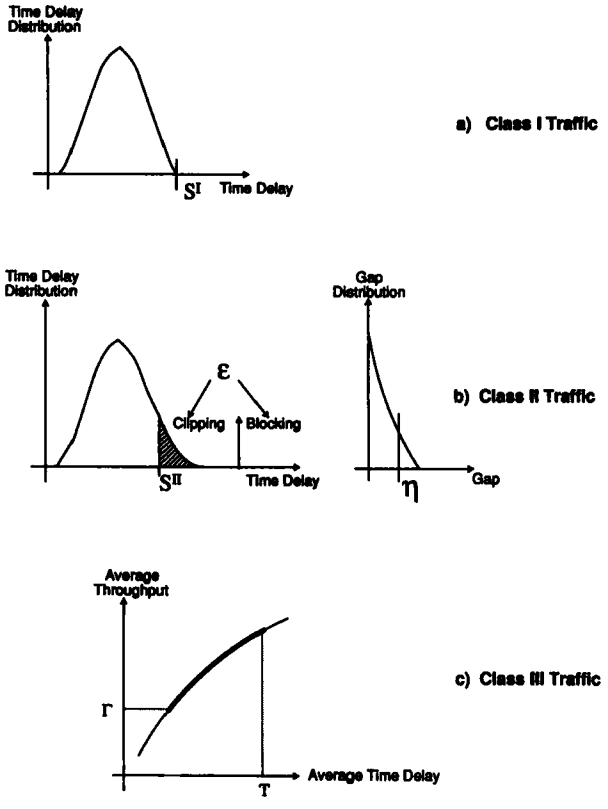


Figure 2. Characterization of the quality of service for user traffic

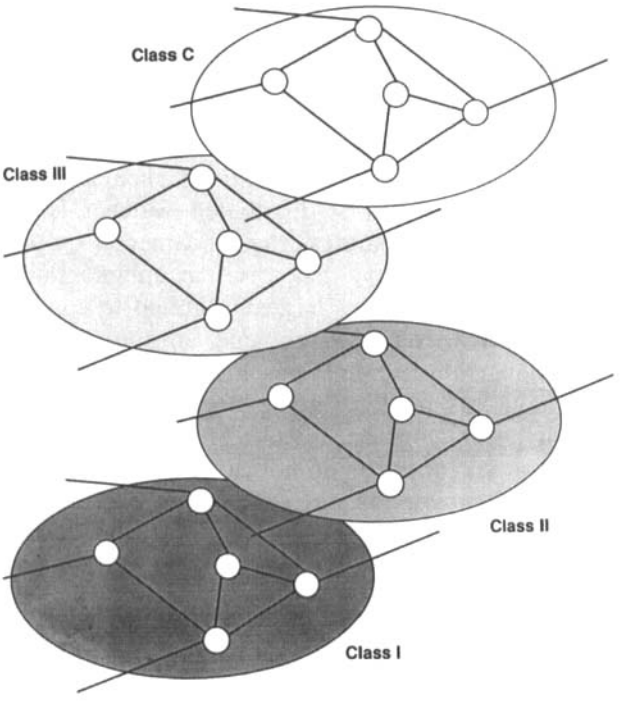


Figure 3. The four virtual networks

by the fairness criteria employed, the objective (or utility) function(s) used and the traffic load and profile. The resource allocation problem is resolved by the TCA by using four types of resource-sharing algorithms: admission control, flow control, routeing and scheduling and buffer management.[7, 12] Queuing analyses that discuss admission control and flow control of real-time packet traffic are given in References 13 and 14, respectively.

In addition to the basic division of traffic into four classes, a *priority* mechanism can be introduced within each class. We propose up to four levels of priority for each Class C, I, II and III traffic. The priority mechanism can be used in several ways. One example is to associate different values of the performance constraints with each priority level. Thus, each level of Class I would be associated with a different S^{I}, and each level of Class II with a different set of S^{II}, ϵ and η values. Priority levels could also be used to differentiate between virtual circuit and datagram-orientated Class III traffic. Therefore, buffer management systems can use priorities as a basis for dropping packets within a given traffic class. Thus, for the class of networks proposed here, selective packet discarding policies[15] can be supported.

3.2. *Network and user performance*

Performance characteristics play a major role in the process of abstracting the integrated reference model.[2] Two performance criteria are considered: *network performance* and *user performance*. Network performance reflects the global behaviour of the network. Statistics for packets of the same traffic class in the entire network are used to calculate the associated performance indicators. The same statistics apply to user performance, but computation for the associated performance indicators is made for *each* user on the network. Furthermore, the perceived performance measures can be formalized in terms of *utility functions* and *costs*, both parametrized by *constraints, class of control strategies,* and the *structure of information* on which the control algorithms are based.[3, 14, 16] The utility functions and constraints considered here are associated with the three user traffic classes.

The utility of the first class of packets is the *probability of blocking* (i.e. the frequency of blocked calls) and both the *maximum* and the *average throughputs*. The constraint is specified for 0 per cent contention packet loss with a maximum acceptable time delay S^{I}.

The utility of the second traffic class is the *probability of blocking* and the *average throughput*. Both are functions of the traffic load of the different traffic classes as well as the resource sharing mechanism employed. The upper bound on the percentage of contention packet loss and on the average number of consecutively lost packets arise as constraints.

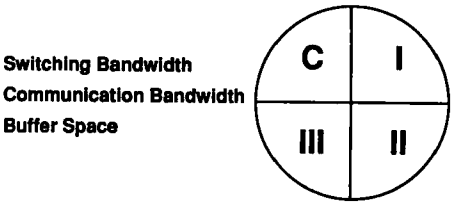
The utility of the third traffic class is characterized by the *average throughput*. The average time delay appears as a constraint and is again parametrized by the traffic load of the different traffic classes and the resource sharing mechanism in use.

3.3. *A general concept for asynchronous resource sharing*

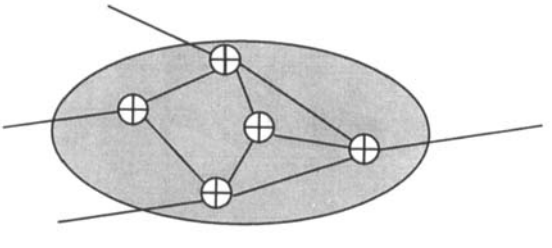
For the multiclass network model described above, scheduling and buffer management resolves contention between the different traffic classes. *Scheduling consists of switching and communication bandwidth allocation, whereas buffer management refers to buffer space partitioning*. The essential requirement on these resource-sharing mechanisms is to guarantee the appropriate quality of service for each traffic class. The quality of service is monitored and controlled by the traffic control architecture (TCA) of the network.[7, 17]

The TCA sees the network as a resource that has to be efficiently allocated among four traffic classes. The pie chart of Figure 4(a) shows the global view of network resources. Our assumption throughout has been that the main network resources: switching bandwidth, communication bandwidth and buffer space, are both observable and controllable (to various degrees). The TCA determines the relative allocation of the above resources to the four classes of service.

The global view of network resource allocation has a distributed implementation. The generic network considered here consists of a set of switching nodes that are interconnected in a mesh topology with high-speed communication links. Each switching node has its own resource allocation. The TCA



a) Global View



b) Distributed View

Figure 4. Network resource allocation

for each switching node finds the position of the boundaries between the Classes C, I, II and III that guarantees the required quality of service (see Figure 4(b)). In the dynamic environment of an integrated network, we envision that these boundaries will be continually changing.

It is the distributed implementation of resource allocation that gives the network architecture its asynchronous nature. For example, at any given point in time, the switching bandwidth of each node in the network could be allocated to any of the four traffic classes. One possible scenario for a five-node network is shown in Figure 5. Each of the nodes is shaded to indicate which traffic class is being served. At the particular time instant shown in the Figure, two nodes are serving Class I traffic, one node is serving Class C, one node is serving Class II and one node is serving Class III. At another time instant, the allocation of nodes to traffic classes could be very different. Although it is not explicitly shown in Figures 4(b) and 5, the same principle applies to the allocation of communication bandwidth for each communication link in the network. The implementation of the general asynchronous resource-sharing principle described above is explained in subsection 4.2.

4. THE BASIC SWITCHING ARCHITECTURE

The architecture proposed here is suitable for a switching node that interconnects a number of high-speed links of an integrated network. The network is assumed to have a mesh topology. Figure 6 shows the topology of the integrated network in which the nodes are embedded. It also shows the basic architecture of a switching node. A switching node contains three basic elements: input buffers, switch fabric and output buffers.

The function of the switching and communication bandwidth schedulers and buffer managers is to implement resource allocation strategies that guarantee the overall quality of service of the switching architecture. The switching bandwidth scheduling mechanism (of the switch fabric) is based on time sharing. Communication bandwidth schedulers (attached to each outgoing link) operate on the same principle. Finally, buffer management is achieved through space partitioning. The basic design principles for each of these algorithms will be described in subsection 4.2.

4.1. *The system model of the switching architecture*

The basic architecture of a switching node is given in Figure 6. The switching node interconnects a set of input communication links with a set of output communication links. It consists of three elements: input buffers, switch fabric and output buffers. The fundamental requirement on the switching architecture is the transfer of information from its inputs to its outputs such that time delay and blocking-sensitive performance criteria are met.

The switching architecture supports four traffic classes. Every access point contains a group of four input buffers, one for each traffic class. Traffic arriving at an access point is stored, according to its class, in one of the four buffers. Each group of four buffers is interconnected to the switch fabric via an input port. (The input port can be modelled as a single server.)

The switch fabric supports the transfer of packets from input buffers to output buffers. (It is usually modelled as a network of queues.) Both single class and multiclass switch fabrics are considered here.

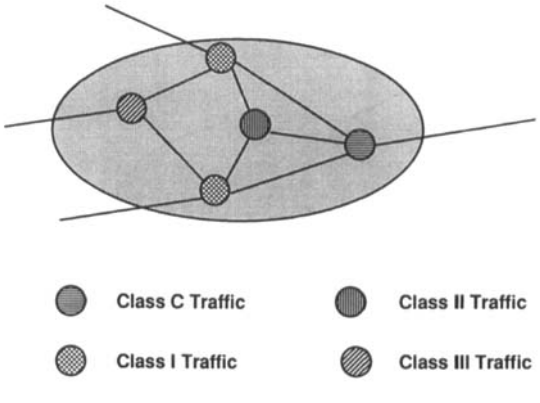


Figure 5. A scenario for switching bandwidth allocation

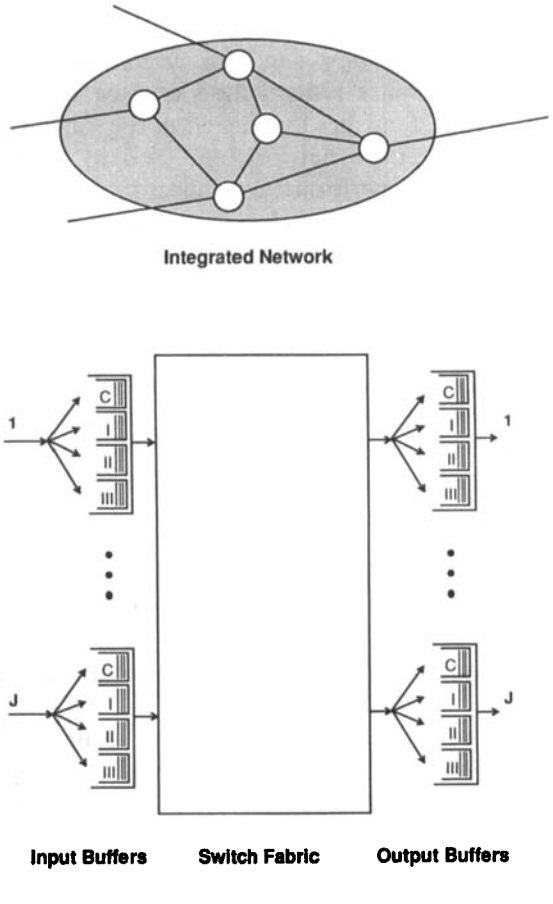


Figure 6. The basic architecture of a switching node

For a single class switch fabric, no distinction is made between different traffic classes by the servers and queues within the fabric. In the multiclass case, the four classes of traffic share the servers inside the switch fabric. Packets, however, are stored within the switch fabric in class-dependent queues. In general, there are two methods of accessing a switch fabric: cell-synchronous and cell-asynchronous. The cell-synchronous mode is typically used with interconnection architectures. In this case, head of the line packets at the various access points that belong to the same traffic class enter the switch fabric simultaneously. Other switch fabrics, for example rings, operate in a cell-asynchronous mode, where packets do not enter the switch fabric at the same time instant. Regardless of the particular implementation, the basic requirement, for the purposes of supporting asynchronous time sharing, is that the switch fabric is non-blocking. That is, the switch fabric transfers information from input to output without packet loss.

There have been a number of switch fabrics proposed in the literature[18–20] for integrated networks. Most of the proposed fabrics are not inherently non-blocking. However, under certain loading or operating conditions this requirement can be satisfied, albeit at the expense of the efficiency of the switch fabric. One possible switch fabric is shown in Figure 7. It consists of two sets of rings that are distributed around a torus. We call this architecture the torus switch fabric.[21] This architecture is particularly appropriate for metropolitan area networks that are typically characterized by low connectivity (see Reference 4 for more details).

The output buffers have the same functionality as the input buffers. Packets exiting the switch fabric are stored in four buffers according to their traffic class. A group of four buffers is interconnected to an outgoing link.

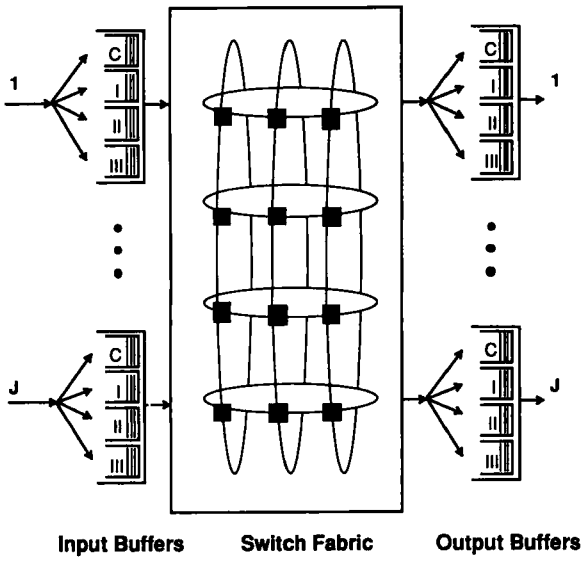


Figure 7. A switching node with a torus switch fabric

4.2. *Asynchronous time sharing*

Asynchronous time sharing (ATS) refers to the manner in which scheduling and buffer management resolves contention between the different traffic classes. ATS calls for dynamic scheduling of the four traffic classes at each contention point in the network. Contention points could arise during the allocation of switching or communication bandwidth, or buffer space.

The basic problem of allocating switching bandwidth is shown in Figure 6. The J access points, each consisting of a group of four queues (one for each traffic class), share a multiple server system (switch fabric). A scheduling policy determines how the servers are allocated among the $J \times 4$ queues. For example, a simple priority scheme might always serve Class C packets first (if available), followed by Class I (if available) followed by Class II (if available) and finally followed by Class III. Since the priority policy always serves Class I traffic before Class II, Class I packets will have a delay smaller than S^{I}, leading to increased Class II contention packet loss. Thus, to satisfy the Class II service requirements, the Class II traffic load must be decreased. Therefore, this type of scheduling policy does not efficiently satisfy the quality of service requirements. A more flexible scheduling policy is needed to provide the appropriate quality of service for each traffic class while operating the network efficiently. The problem of allocating communication bandwidth is similar. In this case, a single access point containing four queues shares a single server system (output link).

The general concept of the proposed scheduling policy for switching and communication bandwidth allocation is shown in Figure 8. The switching (or communication) bandwidth is divided into time periods called cycles. Each cycle is divided into four subcycles. During each subcycle (C, I, II, III), the switch fabric is allocated to the corresponding traffic class (C, I, II, III). For example, during subcycle C, Class C packets enter the switch fabric. The length of a subcycle is measured in cells. A cell represents the time required to serve (switch) one packet. The boundaries between subcycles are determined by a maximum length movable boundary scheme. We consider two different implementations of this scheme, called Mode A and Mode B, which are shown in Figures 8(a) and 8(b).

For Mode A, the TCA uses four variables (MAX C, MAX I, MAX II and MAX III) to determine the maximum boundary positions between subcycles. MAX C represents the maximum length (in cells) of subcycle C. MAX I represents the maximum length of subcycles C and I combined. MAX II represents the maximum length of subcycles C, I and II combined. MAX III represents the maximum length of the entire cycle. These variables are controlled by the TCA of the switch and will dynamically change according to the traffic load and mix. However, MAX C will be fixed and represents

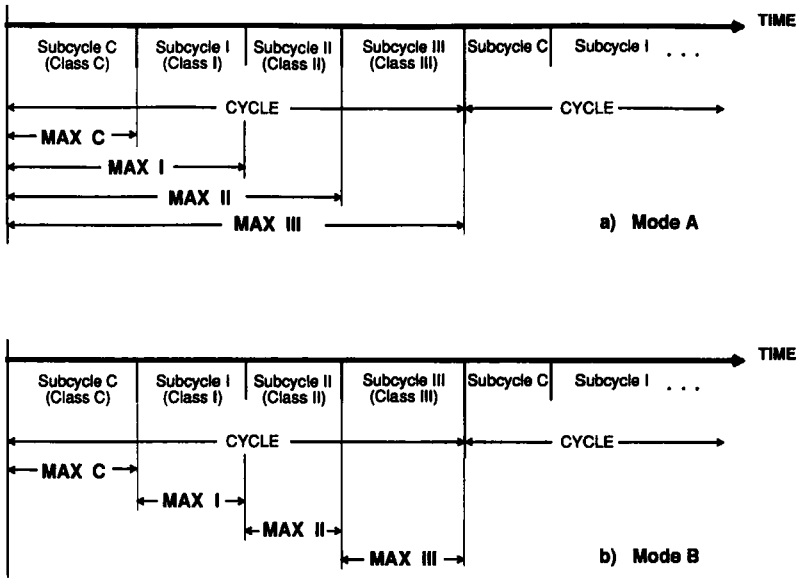


Figure 8. Switching and communication bandwidth allocation

the maximum amount of bandwidth allocated to Class C traffic.

In addition to the maximum length constraint, a movable boundary scheme is used. This method switches subcycles when no more packets of the current traffic class are available. Thus, at the beginning of a cycle, the switch is allocated to Class C. The switch will serve Class C until either MAX C is reached or there are no more Class C packets available. At this point, the switch will change to subcycle I and serve Class I traffic. Class I traffic will be served until MAX I is reached or there are no more Class I packets available. When either condition occurs, the switch will change to subcycle II. When MAX II occurs or there are no more Class II packets available, the switch will start subcycle III. Finally, a new cycle begins when MAX III is reached or there are no more Class III packets available.

Mode B is similar to Mode A, except that a different interpretation of the MAX variables is used. For Mode B, MAX C represents the maximum length of subcycle C, MAX I represents the maximum length of subcycle I, MAX II represents the maximum length of subcycle II and MAX III represents the maximum length of subcycle III. The maximum length of the entire cycle is: MAX C + MAX I + MAX II + MAX III.

The basic difference between the two modes is the manner in which the movable boundary scheme reallocates unused cells to other subcycles. For Mode A, unused cells from one subcycle are made available to the next subcycle in the scheduling sequence. For example, if all available Class C packets are served before MAX C is reached, then the unused bandwidth is allocated to Class I. In this case, the actual number of cells used for Class I could exceed MAX I − MAX C. For Mode B, if a particular subcycle does not use the allocated bandwidth, then the length of the entire cycle is shortened. This causes the scheduler to return to each subcycle sooner than the maximum cycle limit. Thus, Mode A distributes unused bandwidth in a prioritized fashion, whereas Mode B distributes unused bandwidth among all traffic classes.

For a given traffic class, the available bandwidth must be allocated fairly among the multiple access points. A method to limit access in order to guarantee users the appropriate bandwidth is proposed here. Each access point is assigned four *limit* variables (L^C, L^I, L^{II}, L^{III}) by the TCA. These variables are defined as the maximum number of packets of each class that the access point can transmit during one cycle. For example, if the TCA assigns access point X an L^{II} value of 5, then access X can transmit no more than five Class II packets each cycle. To solve the output port congestion problem mentioned in Section 2, the *limit* variables concept can be extended. In its full generality, each access point contains a set of *limit* vectors which defines the number of packets it can send to each output port for each traffic class. For example L^I, the *limit* for Class I traffic assigned to an arbitrary input access point, is a vector of the form $L^I = (L^I(1), L^I(2), \ldots, L^I(J))$. The TCA dynamically controls these variables according to the traffic load and profile.

Each access point to a switch fabric or communication link requires a buffer organization that supports the four traffic classes. This was shown conceptually in Figure 6 as four separate FIFO memories. The total memory space at each access point, however, is considered a common buffer pool for the use of all traffic classes. This pool is divided

into four areas using space partitioning as shown in Figure 9. Each buffer pool is assigned four *threshold* variables (B^C, B^I, B^{II}, B^{III}) by the TCA. A *threshold* variable determines the maximum number of packets of a traffic class that are allowed into the common buffer. Once the *threshold* value is reached, no additional packets of that class are accepted. For example, if the B^{III} value is 7, then no more than 7 Class III packets are allowed into the buffer. The TCA determines the values of these variables using static or dynamic reconfigurability algorithms. In the static case, the variables are set according to the expected average traffic load and profile. In the dynamic case, the variables are continually changing according to the changing traffic load and profile on the network.

In addition to the basic space partitioning among classes, the buffer management system handles the four level priority scheme proposed for each class. Thus, the space assigned to each class is subdivided into four queues which can be accessed independently. The priorities can be used as a basis for dropping packets within a given class. For example, if the *threshold* for a given class has been reached, a new arrival could be allowed into the buffer by dropping a lower priority packet of the same class that is already in the buffer.

The ultimate goal of the TCA is to guarantee quality of service for the different traffic classes in the network. How can this be achieved in a network design based on the principles of ATS? The time/space resource allocation strategy presented above readily allows the evaluation of the performance parameters for Class I, II and III traffic. This is because by controlling the time of occupancy of a resource, a controllable amount of switching (or communication) bandwidth is provided for each traffic class and access point. In addition, by allocating the buffer space, the maximum number of packets stored for a particular traffic class can also be controlled. As a result, the expected delay and loss characteristics of all traffic classes can be evaluated at each switching node. Consequently, the *end-to-end* delay and loss characteristics can be predicted for all traffic classes in the network. Should a new call request admission, the impact of its addition to the network can be predicted based on the current performance parameters and the quality of service descriptor of the new call. If the performance of the already existing users and that of the new user can be guaranteed, the call is accepted; otherwise the call is rejected.[12]

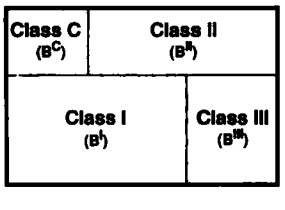


Figure 9. Buffer management

In order to support these capabilities, networks based on the ATS principle require distributed sensors for traffic monitoring and evaluation. Typically, monitored events are buffer occupancy, average throughput, time delay, total packet loss and consecutive packet loss. The sensors are attached to network buffers where these events can be extracted. There is a need for implementing these monitors together with the corresponding buffers compactly in hardware. Finally, because of the fast state and events occurring in integrated networks, traffic evaluation should be done locally as well. Thus, intelligent buffer management systems that can not only monitor but also evaluate the appropriate statistics are needed.

5. CONCLUSIONS

The asynchronous time sharing principle for designing and implementing integrated networks that guarantee quality of service has been presented. This principle requires the core of the network to recognize four traffic classes. These classes have been defined using a set of time-sensitive and blocking-sensitive parameters. A set of switching architectures has been also described that supports asynchronous time sharing. The basic architecture can employ existing switch fabrics with some appropriate modifications. This permits experimentation with many types of fabrics according to the performance requirements under consideration.

In order to evaluate the ATS concepts presented in this paper, we implemented a network test-bed called MAGNET II.[4] The test-bed serves as a platform for developing and evaluating real-time scheduling and buffer management algorithms. It also provides a vehicle for experiments that can evaluate the efficiency of various routeing, flow control and admission control policies that were not covered in this paper. A set of real-time measurements on MAGNET II is presented in Reference 22. Additional results will be published elsewhere.

Classical design of integrated networks starts with the design of the network architecture, followed by the design of the management architecture. This sequence of events creates complexity problems that are very hard to overcome because management primitives do not appear in the network architecture early in the design process. By taking into account requirements of the traffic control architecture in the definition phase of the network architecture (for example traffic classes and intelligent buffer management systems), we hope to have given an example of a new generation of intelligent integrated networks that will better respond to user needs and requirements.

ACKNOWLEDGEMENT

The research reported here was supported by the U.S. National Science Foundation under Grant # CDR-84-21402.

REFERENCES

1. L. Zhang, 'Designing a new architecture for packet switching communication networks', *IEEE Communications Magazine*, **25**, (9), 5–12 (1987).
2. A. A. Lazar, M. A. Mays and K. Hori, 'A reference model for integrated local area networks', *Proc. International Conference on Communications*, Toronto, Canada, 22–25 June 1986, pp. 531–536.
3. A. A. Lazar, A. Patir, T. Takahashi and M. El Zarki, 'MAGNET: Columbia's integrated network testbed', *IEEE J. Selected Areas in Communications*, **SAC-3**, (6), 859–871 (1985).
4. A. A. Lazar, A. Temple and R. Gidron, 'MAGNET II: a metropolitan area network based on asynchronous time sharing', *IEEE J. Selected Areas in Communications*, **SAC-8**, (8), (1990).
5. A. A. Lazar and J. S. White, 'Packetized video on MAGNET', *Optical Engineering*, **26**, (7), 596–602 (1987).
6. M. El Zarki, A. A. Lazar, A. Patir and T. Takahashi, 'Performance evaluation of MAGNET protocols', in R. L. Pickholtz (ed.) *Local Area and Multiple Access Networks*, Computer Science Press, 1986, pp. 137–154.
7. A. A. Lazar, 'Object-oriented modeling of the architecture of integrated networks', *CTR Technical Report # 167-90-04*, Center for Telecommunications Research, Columbia University, New York, January 1990.
8. J. M. Ferrandiz and A. A. Lazar, 'Consecutive packet loss in real-time packet traffic', *Proc. Fourth International Conference on Data Communication Systems and their Performance*, Barcelona, Spain, 20–22 June 1990.
9. T. O. Brunner and J. S. White, 'Implementation of packet telephone and video services on a local area network', *CTR Technical Report # 103-88-42*, Center for Telecommunications Research, Columbia University, August 1988.
10. B. Maglaris, D. Anastassiou, P. Sen, G. Karlsson and J. D. Robbins, 'Performance models of statistical multiplexing in packet video communications', *IEEE Trans. Communications*, **COM-36**, (7), 834–844 (1988).
11. G. Karlsson and M. Vetterli, 'Subband coding of video for packet networks', *Optical Engineering*, **27**, (7), 574–586 (1988).
12. A. A. Lazar, 'The game of networking', *CTR Technical Report # 200-90-37*, Center for Telecommunications Research, Columbia University, New York, July 1990.
13. J. M. Ferrandiz and A. A. Lazar, 'Admission control for real-time packet sessions', submitted for publication to *IEEE Trans. Automatic Control*.
14. F. Vakil, M. T. Hsiao and A. A. Lazar, 'Flow control in integrated local area networks', *Performance Evaluation*, **7**, (1), 43–57 (1987).
15. N. Yin, S. Li and T. Stern, 'Congestion control for packet voice', *CTR Technical Report # 78-88-06*, Center for Telecommunications Research, Columbia University, New York, 1988.
16. S. Q. Li and M. El Zarki, 'Dynamic bandwidth allocation on a slotted ring with integrated services', *IEEE Trans. Communications*, **COM-36**, (7), 826–833 (1988).
17. A. A. Lazar, J. T. Amenyo and S. Mazumdar, 'WIENER: a distributed expert system for dynamic resource allocation in integrated networks', *Proc. IEEE Symposium on Intelligent Control*, Philadelphia, PA, 18–20 January 1987, pp. 159–164.
18. P. Gonet, P. Adam and J. P. Coudreuse, 'Asynchronous-time-division switching: the way to flexible broadband communication networks', *Proc. International Zürich Seminar on Digital Communications*, Zürich, Switzerland, 11–13 March 1986, pp. 141–148.
19. A. Huang and S. Knauer, 'Starlite: a wideband digital switch', *Proc. IEEE Global Telecommunications Conference*, Atlanta, GA, 26–29 November 1984, pp. 5.3.1–5.3.5.
20. J. S. Turner, 'New directions in communications', *Proc. 1986 International Zürich Seminar on Digital Communications*, Zürich, Switzerland, 11–13 March 1986, pp. 25–32.
21. von C. Conta, 'Torus and other networks as communication networks with up to some hundred points', *IEEE Trans. Computers*, **C-32**, (7), 657–666 (1983).
22. A. A. Lazar, G. Pacifici and J. S. White, 'Real-time traffic measurements on MAGNET II', *IEEE J. Selected Areas in Communications*, **SAC-8**, (3), 467–483 (1990).

Authors' biographies:

Aurel A. Lazar was born in Zalau, Transylvania, Romania, on 30 January 1950. He received the Dipl.-Ing. degree in communications engineering (Nachrichtentechnik) from the Technische Hochschule Darmstadt, Darmstadt, Federal Republic of Germany, in 1976, and the Ph.D. degree in information sciences and systems from Princeton University, Princeton, NJ, in 1980. In 1980 he joined the faculty in the Department of Electrical Engineering of Columbia University as an Assistant Professor. Since 1988 he has been a Professor and Director of the Telecommunication Networks Laboratory. He is an editor of the Springer Verlag monograph series on Telecommunication Networks and Computer Systems and editor for Voice/Data Networks of the *IEEE Transactions on Communications*. He is a founding member of the Center for Telecommunications Research at Columbia University, a member of IEEE and ACM. His current areas of interest include control and management of telecommunication networks and, the mathematics of networks and intelligent systems.

Adam T. Temple received the B.S. degree in engineering from Yale University in 1979 and the M.S.E.E. degree from Columbia University in 1987. From 1979 to 1985, he designed high-speed signal processing systems for sonar applications at Raytheon's Submarine Signal Division. Since 1985, he has been a member of the research staff at the Center for Telecommunications Research at Columbia University. His major technical activities there have been in the areas of broadband integrated networks and computer communication networks.

Rafael Gidron was born in Tel Aviv, Israel, on 3 March 1958. He received the B.S. degree (Magna Cum Laude) from Tel Aviv University in 1985, the M.S. degree from Columbia University in 1987, and is currently in the Ph.D. program at Columbia University. Mr Gidron has been a research staff member at the Center for Telecommunications Research since 1985, working in system and hardware design of broadband integrated networks. Before joining the CTR, he has been working with the Israeli Defence Forces, Tel Aviv University and Simteck in Israel and with Renecentrallen in Denmark. His current interests include hardware architecture and resource allocation in broadband integrated networks, as well as digital signal processing and multiprocessing architecture.