A Layered Broadband Switching Architecture with Physical or Virtual Path Configurations

Joseph Y. Hui, *Senior Member, IEEE,* Melike B. Gursoy, Nader Moayeri, *Senior Member, IEEE,* and Roy D. Yates, *Member, IEEE*

***Abstract*—We describe a multilayer connection control architecture for broadband communications. A graph framework is introduced to describe network layers of network design, path configurations, dynamic call routing, burst switching, and ATM cell switching. These hierarchical layers of switching are performed at decreasing time scales, respectively. Switching at the higher layer is performed to reduce blocking at the next smaller time scale. A layered notion of equivalent bandwidth for satisfying layered grade-of-service parameters is introduced for making connections at these time scales. We then focus on the path configuration layer. Two path setup methods, namely physical and virtual path setup, are described. Mathematical programs minimizing path bandwidth usage subject to meeting grade-of-service requirements are formulated for both methods. The relative merits of these methods are compared. In one example, physical path setup is shown to require roughly 50% more bandwidth than virtual path setup.**

I. Introduction

ONE major distinction of integrated broadband networks from telephony is the heterogeneity of communication connections. Such heterogeneity results from their dramatically different traffic characteristics. First, call sessions between end terminals may have a broad spectrum of holding times and bit rate requirements. Second, the bit stream produced per session is often bursty in character. Third, call sessions between two points in the network could be aggregated and transported over a preassigned path and, hence, beyond the call layer, connections also have to be managed at the path layer.

To handle traffic heterogeneity, new communication standards are being considered for adoption for integrated broadband networks. The asynchronous transfer mode (ATM) [1], [2] is the preferred multiplexing format for switching and transporting multirate and bursty call sessions. Each call is assigned a virtual circuit. A burst within a call session is considered as a packet, which is further fragmented into ATM cells via an adaptation layer [3]. Beyond managing sessions, a network may assign a path between two nodes in the network [4]–[6]. A virtual circuit could be established using one or more connected paths.

Manuscript received January 15, 1991; revised August 6, 1991. The work of J. Y. Hui was supported by the National Science Foundation under contract PYI NCR 90-58079, by Bell Communications Research, and the NEC Research Institute.

The authors are with Rutgers University, Piscataway, NJ 08855-0909.

IEEE Log Number 9103924.

To facilitate transport at these layers, various switching mechanisms are proposed. These mechanisms are distinguished by the time scales of the connections established. ATM cells, which last a few microseconds, are switched by ATM switches [7], [8]. Bursts or packets, which last typically less than 1 second, could be switched by a form of burst switching [9]–[12]. Call sessions, which last for minutes, are switched by assigning a virtual circuit established on a sequence of paths [4]–[6]. For dynamic call routing [13]–[15], alternative path sequences may be considered according to their degree of congestion. Paths, which last for hours or days depending on how fast the aggregate traffic fluctuates, are switched and configured by either ATM cross connects [5], [6] or facility cross connects [4], [13]. A more detailed description of these switching mechanisms are described in the next section.

Switching is layered by time scales [9]–[11], [16], [17] primarily for the following technological and traffic concerns. The first concern is the relative use of optical versus electronic switching technologies. Compared with electronic switching, optical switching technologies have a higher information throughput in a fixed switch setting but a lower reconfiguration rate for the switch setting. Subsequently, optical switching is best suited for the higher levels of the network, where the substantially aggregated traffic has high volume but less frequent fluctuations due to statistical averaging. The second concern is the reduction of computation required to set up a connection. By shifting functions to switching at a larger time scale, we avoid performing such functions repetitively at a smaller time scale. For example, the function of routing is performed per packet for datagram communications, which could be avoided at the packet layer if a virtual circuit layer is present. The third concern is the need for abstraction in the process of resource allocation. Abstraction of resource allocation processes provided by lower layer protocols simplifies communication between peer protocols and peer networks.

Besides a formal description of network layers for heterogeneous services, the major contribution of this paper is to structure bandwidth management for heterogeneous service via a layered notion of equivalent bandwidth. The equivalent bandwidth is requested to satisfy the grade of service (GOS) up to the layer of concern. The most important use of this layered equivalent bandwidth is for

0733-8716/91$01.00 © 1991 IEEE

bandwidth allocation at the path layer. Optimization algorithms are given for models of physical or virtual path networks. Consequently, we are able to quantitatively compare the efficiency for these two path configuration methods.

The paper is organized as follows. Section II describes the layered switching architecture. A grammar for generating network layers is described. Using this grammar and layering framework, we describe the layers encountered in practice, including the facility network, the physical/virtual path network, the call routing network, burst switching, and ATM cell switching. Section III explores how adjacent layers of this switching architecture interact. A principle for trading off switching between two adjacent layers is described. A layered notion of equivalent bandwidth for this tradeoff is introduced from the cell layer up to the call layer. Sections IV and V focus on the mathematical formulation of physical and virtual path configurations, respectively. Section VI compares the relative merit of physical versus virtual path reconfigurations and describes an object-oriented graphical simulation/design software developed for the switching architecture described in this paper.

II. Architecture for Network Layering

In a broad sense, switching is the process of allocating transmission resources and the transfer of information along one or more paths formed by the resources allocated. Blocking is the event that such allocation and transfer fail. Consequently, allowing a large set of alternative resources for switching would reduce the likelihood of blocking. However, these alternatives also increase the complexity of switching. Therefore, a tradeoff is necessary between reducing blocking and reducing processing complexity.

One method to reduce complexity is by hiding complexity through abstraction. Abstraction is the process of simplifying the representation of sets of resource or traffic entities. By layering switching according to time scales, switching at each layer only has to manage an abstracted view of the resources allocated and traffic states at that time scale.

Switching at one time scale involves the adoption of a subset of alternatives adopted by switching at the next larger time scale. The choice of the subset takes into account the events of blocking which happen at the next smaller time scale. Subsequently, the complexity due to size of the subset has to be balanced against the resulting probability of blocking. Later, we shall state more precisely this principle of layered switching, and illustrate the principle using commonly known practices of switching.

Before describing the layered switching architecture, we introduce an iterative grammar for generating a network from another network. Suppose we have a graph $G_i = (N, E_i)$ in which i represents the iterate index (starting with $i = 0$), N is the set of nodes in a communication network, and $E_i \subset N \times N$ is a set of links between nodes. We now iteratively define a graph G_{i+1} in the following manner. Let $\omega = (o, d)$ be an origin–destination pair with $o, d \in N$. let p_ω be a path in G_i; namely, an acyclic sequence of adjacently connected nodes from o to d. Let $P_\omega = \{p_\omega\}$ be a path group; namely a subset of all possible paths between ω. We define $G_{i+1} = (N, E_{i+1})$, in which E_{i+1} consists of edges ω with nonempty P_ω. In essence, each edge in G_{i+1} is an abstraction of the means P_ω for facilitating connections for ω.

We now use this graph framework to describe a layered switching architecture, which is illustrated in Fig. 1. Some of the following layers are present in the telephone network. This paper reexamines some practices of telephony in the context of broadband services, which is nonhomogeneous and may require more layering than telephony.

To start with, facility deployment involves the design of the graph G_0; namely the facility links between the nodes as shown in Fig. 1. The nodes in the facility network are not necessarily directly connected. To enhance apparent direct connectedness, paths (solid lines connecting nodes) between nodes are configured as shown. The paths are configured by facility cross-connect switches, which makes the connection within each node (marked by solid lines within each square in Fig. 1. The subsequent G_1 graph is termed the path network, which is shown in Fig. 2. Each edge in G_1 is termed a path group, consisting of a set of paths in G_0. Notice that there can be more than one path between two nodes (for example, the path (b, d) and the path (b, c, d) for connecting the nodes b and d). Besides enhancing the apparent connectedness of nodes, preconfigured paths also simplify the process of call routing. Also, path reconfigurations allow adaptation to traffic changes or link/node failures.

These paths can be configured physically or virtually. For physical path configurations, we assume that each path p is allocated, without sharing with other paths, a capacity f_p from each edge in the path p in the facility network G_0. We assume that each edge $e \in E_0$ has a capacity C_e. Let Q_e be the set of paths which use the edge e. Consequently, the sum of capacities assigned to paths in Q_e must be less than its capacity C_e.

Instead of physical bandwidth allocation, the allocation can be virtual in the sense that the paths may share the bandwidth dynamically without a rigid assignment of bandwidth to each path. The path group in G_1 is called a virtual path group P_ω, which consists of a set of virtual paths p_ω. Each link of a virtual path is given a virtual path identifier (VPI). ATM cross-connect switches are used to configure these virtual paths at the nodes. These switches examine the VPI of each ATM cell to route the cell to the next link in the virtual path. Each ATM cross connect maps the VPI of a cell into a physical switch output address for the purpose of routing within the switch fabric. Also, the VPI is translated into the VPI for the next link in the path. At the end of a virtual path, the ATM cell either arrives at its destination node, or is transferred to another virtual path in the route used by a virtual circuit.

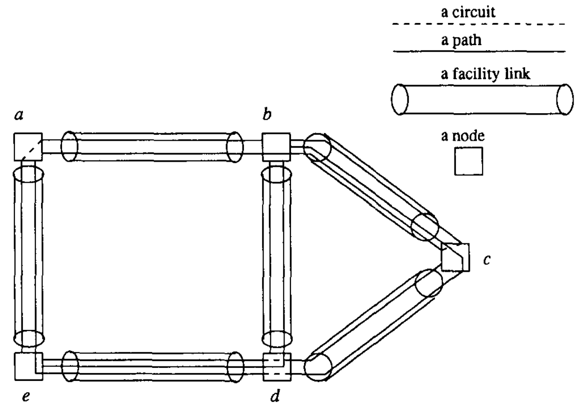


Fig. 1. Layered connection management.

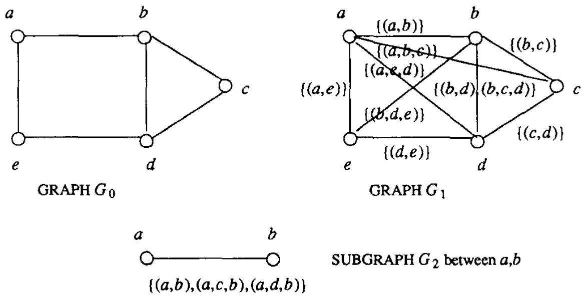


Fig. 2. The networks G_0, G_1, G_2.

During call session setup, alternate routing can be facilitated in G_1 by choosing a physical or virtual path in the path group P_ω. However, the path diversity provided by the path group P_ω is limited. To achieve a small probability of call blocking, the path group P_ω has to be substantially oversized. This oversizing is more serious for physical paths because bandwidth is dedicated. Also, the path group in G_1 may not have a large number of paths, and hence insufficient path diversity is provided at the path layer to handle traffic volatility or link failure.

To enhance path diversity, a further iterate on G_1 to generate an alternate routing network G_2 may suffice. A route in G_2 is a sequence of tandem path groups (physical or virtual) in G_1. An edge in G_2, termed a route set, is a set of alternate routes between the nodes link by the edge. As shown in Fig. 1, a circuit between the nodes b and e can be established using two routes: the direct path (b, d, e) or the route consisting of the paths (b, a) and (a, e). Even though these two routes traverse the same number of links in the facility network, they are switched by different kinds of switches at the intermediate node. As shown in Fig. 1 for each node, path connections facilitated by cross-connect switches are marked by solid lines, whereas circuits facilitated by call switches are marked by dotted lines. A circuit which employs a multiple-path route may visit the same node twice (for example, the multiple-path route formed by the path (a, b, c) and the path (c, b) as shown in Fig. 1 for establishing a circuit between nodes a and b). This circuitous arrangement may become necessary if the direct path (a, b) is overloaded. The same circuit could also be established by the tandem paths consisting of the path (a, e, d) and the path (d, c, b). In Fig. 2, the route set for three routes between nodes a and b is illustrated.

For virtual path networks, using an indirect route may require the translation per cell of its VPI–VCI pair at the tandem node into a new VPI–VCI pair for the outgoing tandem path.

The route set may be ordered for call setup considerations. An arriving call may first consider the direct path group (the route that uses a single G_1 edge between two nodes). If the direct path group is congested, other alternate routes are considered in the prescribed order. The alternate routing network G_2 (the routes and perhaps their order) could be reconfigured from hour to hour for changing traffic patterns. The design of the routing network provides more routing alternatives besides the direct path group, while at the same time ignores the not so viable alternatives in order to reduce the burden of computation.

However, the derivation of the route set and its order for each edge in G_2 is computationally tedious and may require off-line computation [13]. Alternatively, a weight (a function of the congestion level) may be assigned to each route in the route set. An arriving call may choose the route with the smallest weight [14]. The congestion level could be updated every few seconds. The resulting routing method is often more adaptive than the previous method of route planning. The alternate routing network layer consists simply of all possible acyclic path sequences in the path network which could be used for call setup. In this case, the route network G_2 is there though there is no need for an elaborate description.

Having described the network configuration procedures at the facility, path, and route layers, we now proceed to describe the allocation procedures at the call, burst, and cell layers. The relationship between these configuration and allocation procedures are illustrated in Fig. 3.

Dynamic call routing is the process of selecting one of the alternate routes for setting up a circuit as described before. For nonbursty calls, the selection of a path within a path group could be done during call setup. For bursty calls, the selection of a path within a path group may be left for the burst layer protocol.

There can be several forms of burst switching. First, a burst can be viewed as a call session by itself, and hence there is no need for distinguishing the burst layer and the call layer. For connectionless service, burst routing is set up per burst without the call layer. However, the per burst processing can be excessive for communication calls with multiple bursts, since a large number of the functions could be collectively performed at the call layer. Assuming that a route (either the direct path group or an alternate path group sequence) is allocated during virtual call setup, burst switching facilitates a more refined choice of the resources allocated, such as by choosing a particular path within each path group of the path group sequence estab-

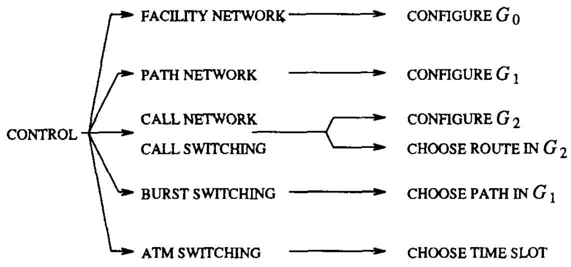


Fig. 3. The resource allocation process.

lished during call setup. The peak bit rate (the inverse of the cell interarrival period multiplied by the channel rate) for a burst is declared at the beginning of a burst. A fast reservation protocol (FRP) [9], [11] then reserves the peak bit rate in a path with sufficient capacity until the burst completes its passage through the path. This form of burst switching achieves three functions: first, the balancing of load within a path group; second, the elimination of out-of-sequence cells caused by different delays in different paths; and third, the blocking of a burst if accepting the burst may deteriorate cell loss probability within a path group.

At the smallest time scale, ATM cell switching switches and transports ATM cells by allocating time slots via some form of queueing discipline (input, internal, or output buffered, FCFS, etc.) The queueing discipline is specific for each switch fabric design. The resource allocation at this time scale is mostly assigned in the time domain, rather than the space domain for larger time scales.

III. Layered Switching Using Equivalent Bandwidth

Having described the various layers of connection management, we now examine how a layer interacts with adjacent layers. We shall put this problem in the context of resource allocation [6], [9]–[11], [17]–[19]. The key idea in this paper is that switching at a larger time scale should attempt to provide a certain grade-of-service (GOS) guarantee for layer at the smaller time scale. In this paper, GOS is measured largely by the amount of blocking, which is quantified for each layer. The general principle we adopt as a guideline for switching at a layer is as follows.

Principle of Layered Switching: Allocate sufficient transport bandwidth for traffic entities at a layer so as to bound the amount of blocking occurring at the next lower layer.

Here, traffic entities can be paths, calls, bursts, or cells. To clarify the terminology, we apply this principle to the layered architecture described in the previous section. First, the facility network should be designed so that the probability of failure to set up a path is small. A path should be set up so that the probability of failure to use the direct path for setting up a call is small. An alternate routing network should be set up so that the probability of call blocking, as a consequence of congestion on the direct path group and the alternate routes, is small. A call should be routed so that the probability of failure to route a burst within a setup route is small. A burst should be routed through a less congested path within a path group so that the probability of cell loss is small.

These blocking probabilities shall be regarded as measures of grade-of-service (GOS). Here, we emphasize that multiple layered GOS requirements should be maintained simultaneously. In communication analysis, GOS is often calculated as a function of traffic statistics and the bandwidth allocated for the traffic. In this paper, we focus on the inverse problem; namely, finding the amount of bandwidth required to transport a certain amount of traffic while maintaining certain GOS requirements. Therefore, the switching control at a layer computes and tries to allocate this bandwidth, termed equivalent bandwidth, for satisfactory transport of traffic entities.

Fig. 4 illustrates the traffic process for the path, call, burst, and cell layers for a single service type. Let λ_l denote the arrival rate of the traffic entity at the layer l. Relative to layer l, we define layers $l+$ and $l-$ as the layers with the next larger time scale and the next smaller time scale, respectively. At the path layer, λ_{path} is the sequence of call arrival rates $\lambda_{\text{call},t}$ over multiple discrete time periods t for an o–d pair. At the call layer, λ_{call} is the call arrival rate for a traffic period. At the burst layer, λ_{burst} is the burst arrival rate for a completed call. At the cell layer, λ_{cell} is the cell arrival rate for a completed burst. Let $1/\mu_l$ be the mean holding time for the traffic entity at the layer of concern. Multiple service types can be represented by different values for these parameters as well as a description of the interarrival and holding time distributions. Constant bit rate (CBR) call types may have the burst layer missing. We underscore a parameter to represent a vector of such parameters for different service types. For example, $\underline{\lambda}_{\text{call}}$ is the vector of call arrival rates in a given traffic period for different service types.

The state of a bandwidth resource at a layer is defined as how the resource is utilized at the time scale of concern. At the cell time scale, the state of an ATM queue could be described by N_{cell}; namely, the number of cells in the queue. At the burst time scale, the state of a path is given by $\underline{N}_{\text{burst}}$; namely, the vector of the number of admitted bursts for different service types. The state of a path group at the call layer may be represented by $\underline{N}_{\text{call}}$; namely, the vector of the number of admitted calls of different types. At the path time scale, $\underline{N}_{\text{path}}$ is the vector of call rates for which paths are engineered or configured to accommodate.

In the set of all possible states $\underline{N}_l$, an admission policy [24] at a layer defines a subset Ω_l of admissible states. An arriving traffic entity may be admitted if the subsequently incremented $\underline{N}_l$ belongs to Ω_l, otherwise it is blocked by the bandwidth resource manager applying the admission policy at that layer. We shall assume coordinate convex policies [24]; that is, for each $\underline{N}_l \in \Omega_l$, decreasing the value of any one of the component $\underline{N}_l$ still makes the resulting $\underline{N}_l$ a member of Ω_l. A coordinate convex policy is illus-

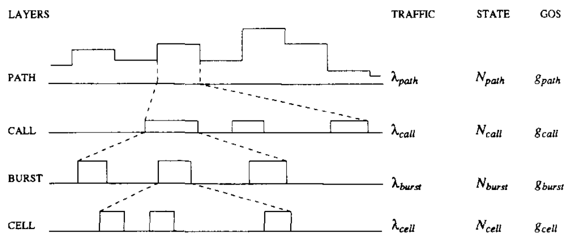


Fig. 4. Time-scaled traffic entities.

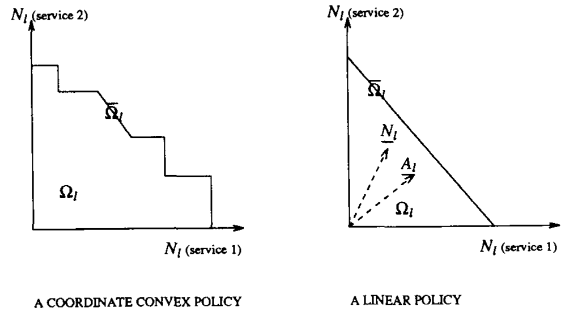


Fig. 5. Admission policies at a layer.

trated in Fig. 5. The set of blocking states $\overline{\Omega}_l$ is defined as the boundary subset of Ω_l for which an arrival (incrementing a component of $\underline{N}_l$) is blocked since the subsequent $\underline{N}_l$ is not in Ω_l.

An admission policy is linear (fig. 5) if all $\underline{N}_l \in \Omega_l$ satisfy the scalar product constraint

$$\underline{A}_l \underline{N}_l \leq 1 \tag{1}$$

in which $\underline{A}_l$ is a constant vector with all components strictly positive. Furthermore, $\underline{A}_l$ is normal to the boundary $\overline{\Omega}_l$. Some preliminary results have shown that linear policies are close to being optimal for high-speed networks and are simple for blocking probability evaluation.

The blocking probability at a layer is defined as

$$B_l = P(\overline{\Omega}_l). \tag{2}$$

For the sake of simplicity, we prefer this notion of time congestion (the fraction of time the system is in a congested state) to call congestion (the fraction of traffic entities blocked, which assumes different values for different service types).

Computing B_l requires more details concerning the stochastic process $\underline{N}_l(t)$. This computation is complicated by the fact that the state transition dynamics at a layer are controlled or modulated by the resource allocation process at the layer of the next larger time scale. In fact, the aggregate arrival rate at a layer for each service type is given by $N_{l+}\lambda_l$ (ignoring the subscript for the service type).

To simplify the analysis, we shall assume that state transitions at the higher layers are quasistatic compared to state transitions at the layer of concern. This assumption is justified by the fact that the time scales for two adjacent layers often differ by two or more orders of magnitude. Given the quasistatic state $\underline{N}_{l+}$, we shall assume state transition rates for $\underline{N}_l$ as constants. Subsequently, B_l may be computed for the resulting stationary finite state stochastic process.

An admission policy Ω_l is termed satisfactory at that layer if $B_l = P(\overline{\Omega}_l) < g_l$, for a given bound g_l on blocking probability at that layer. Let S_l be the set of satisfactory (and perhaps linear) policies. The optimal policy in S_l minimizes certain costs of implementing the policy. We focus on minimizing the equivalent bandwidth, which is defined recursively from layer to layer.

Assuming a quasistatic state $\underline{N}_l$ at layer l, suppose we know how to compute the equivalent bandwidth $\hat{C}_{l-}(\underline{N}_l)$ which is the required bandwidth at layer $l-$ for satisfying the GOS requirements for all layers with time scales smaller than l. We now want to compute $\hat{C}_l(\underline{N}_{l+})$ for a satisfactory Ω_l (which meets the GOS requirement at layer l) assuming quasistatic state transitions at the layer $l+$. This is given by:

$$\hat{C}_l(\underline{N}_{l+} \mid \Omega_l) = \max_{\underline{N}_l \in \Omega_l} \hat{C}_{l-}(\underline{N}_l) \tag{3}$$

which is the worst-case required bandwidth for the satisfactory policy at layer l. Among all satisfactory policies, the optimal policy is one which minimizes this worst-case required bandwidth, namely

$$\hat{C}_l(\underline{N}_{l+}) = \min_{\Omega_l \in S_l} \hat{C}_l(\underline{N}_{l+} \mid \Omega_l) \tag{4}$$

where S_l is the set of all satisfactory admission policies at a layer. (For notational simplicity, we have omitted the dependence of S_l on N_{l+}.) To simplify the computations, S_l may be restricted to linear satisfactory policies.

Combining the min-max computation, we have the following recursive formula for computing equivalent bandwidth.

$$\hat{C}_l(\underline{N}_{l+}) = \min_{\Omega_l \in S_l} \max_{\underline{N}_l \in \Omega_l} \hat{C}_{l-}(\underline{N}_l). \tag{5}$$

This formula has many interesting mathematical properties. These properties and efficient computation methods for equivalent bandwidth based on the linear policies for multiple service types are discussed in an upcoming paper [20]. In the remainder of this section, we explain the computation using the example of the transport of a still-image browsing service, which we shall use in Section VI for computing path configurations.

The traffic for this service is modeled by an m-layer Poisson modulated Poisson process $(\text{PM})^{m-1}$PP. Specifically, we consider $m = 3$, with Poisson call arrivals modulating Poisson burst arrivals which in turn modulate Poisson cell arrivals. For the sake of concreteness, we provide the following numerical example. Each image is coded by 10 Mb. The transport of an image is rate-controlled at 10 Mb/s so the image transmission time, and consequently the transport delay, is approximately 1 second. Assuming a 48 byte information payload per cell, this peak rate of 10 Mb/s is equivalent to roughly $\lambda_{\text{cell}} =$

26 000 cells per second. The cell holding time $1/\mu_{\text{cell}}$ depends on the speed of the channel, which for 150 Mb/s channels give $1/\mu_{\text{cell}} = 3$ μs. We assume $g_{\text{cell}} = 10^{-7}$. At the burst layer, we assume $\lambda_{\text{burst}} = 0.1$ burst/second which means that, on the average, one image is sent every 10 seconds. The burst holding time is $1/\mu_{\text{burst}} = 1$ second as mentioned earlier. We assume $g_{\text{burst}} = 10^{-5}$. At the call layer, we assume a call arrival rate of $\lambda_{\text{call}} = 0.25$ calls/second and a call holding time of $1/\mu_{\text{call}} = 200$ seconds. Consequently, there are (on the average) $\rho_{\text{call}} = \lambda_{\text{call}}/\mu_{\text{call}} = 50$ calls in progress. We assume $g_{\text{call}} = 10^{-3}$.

We now calculate the equivalent bandwidths at the cell, burst, and call layers for this single-service type. Since we consider only one service type, Ω_l is one-dimensional and is simply characterized by an upper bound $\overline{N}_l$ on the states $N_l \in \Omega_l$. The maximum in (3), (5) is therefore achieved by $N_l = \overline{N}_l$.

At the cell layer, the state of an ATM queue (input or output) depends on the queueing discipline and the ATM switching architecture. In general, a discrete-time Markov chain is used to model N_{cell}. The blocking state is given by the buffer size $\overline{N}_{\text{cell}}$, with blocking probability $B_{\text{cell}} = P(\overline{\Omega}_{\text{cell}}) = P(N_{\text{cell}} = \overline{N}_{\text{cell}})$. Let $\overline{\rho} < 1$ be the maximum utilization level such that $B_{\text{cell}} < g_{\text{cell}}$. We can now define the equivalent bandwidth at the cell layer assuming that the state of the burst layer N_{burst} is quasistatic

$$\hat{C}_{\text{cell}}(N_{\text{burst}}) = \gamma\lambda_{\text{cell}}N_{\text{burst}} \qquad \text{where } \gamma = 1/\overline{\rho}. \tag{6}$$

In essence, a penalty factor γ is incurred on the aggregate peak rate ($N_{\text{burst}}\lambda_{\text{cell}}$) for reducing cell loss due to ATM queueing to below the GOS bound g_{cell}. For output-queued ATM switches [21], $\overline{\rho}$ can be as high as 0.9 for a reasonable buffer size of 40 or more cells. Hence, fragmenting traffic into cells imposes a roughly 10% penalty due to limiting cell loss, in addition to the 10% penalty due to label overhead (a 5 byte header in a 53 byte cell). More generally, this penalty should be quite service-independent.

At the burst layer, we have to maintain a burst blocking probability bound $B_{\text{burst}} < g_{\text{burst}}$ on top of the cell blocking probability bound (achieved with penalty γ). At the burst layer, we assume a quasistatic N_{call}. Hence, the burst layer load is given by $N_{\text{call}}\lambda_{\text{burst}}/\mu_{\text{burst}}$. For a satisfactory Ω_{burst} defined by $N_{\text{burst}} \le \overline{N}_{\text{burst}}$, we have:

$$B_{\text{burst}} = P(\overline{\Omega}_{\text{burst}}) = E\left(\overline{N}_{\text{burst}}, N_{\text{call}}\frac{\lambda_{\text{burst}}}{\mu_{\text{burst}}}\right) < g_{\text{burst}}$$
$$\text{where } E(C, \rho) = \frac{\rho^C/C!}{\sum_{i=0}^{C}\rho^i/i!} \tag{7}$$

is the well-known Erlang blocking formula. The smallest satisfactory Ω_{burst} achieving the minimum in (4), (5) has $\overline{N}_{\text{burst}} = \overline{N}^*_{\text{burst}}$ such that $B_{\text{burst}} = g_{\text{burst}}$ for (7). Since the Erlang blocking formula is defined only for integer values of N_{burst}, we use linear interpolation to find the nonlinear value of $\overline{N}^*_{\text{burst}}$ between the two values of $\overline{N}_{\text{burst}}$ with B_{burst} just above the below g_{burst}. Consequently, we have:

$$\overline{N}^*_{\text{burst}} = E^{-1}\left(g_{\text{burst}}, N_{\text{call}}\frac{\lambda_{\text{burst}}}{\mu_{\text{burst}}}\right) \tag{8}$$

in which $C = E^{-1}(B, \rho)$ is the inverse of the Erlang blocking formula $B = E(C, \rho)$. Obviously, the state $N_{\text{burst}} = \overline{N}^*_{\text{burst}}$ requires the largest cell layer equivalent bandwidth (the maximum in (3), (5) for maintaining g_{cell}). Using (5), (6), (8), the burst layer equivalent bandwidth for maintaining g_{burst} and g_{cell} is given by:

$$\hat{C}_{\text{burst}}(N_{\text{call}}) = \gamma\lambda_{\text{cell}}\overline{N}^*_{\text{burst}}$$
$$= \gamma\lambda_{\text{cell}}E^{-1}\left(g_{\text{burst}}, N_{\text{call}}\frac{\lambda_{\text{burst}}}{\mu_{\text{burst}}}\right). \tag{9}$$

This equivalent bandwidth has the same dimension as λ_{cell}, which can be either in units of cells per time slot or bits per second.

At the call layer, we have to maintain a call blocking probability bound on top of the burst and cell blocking probability bounds. We also assume a quasistatic λ_{call}. The formula for the call layer equivalent bandwidth can be derived in a similar manner to that for the burst layer equivalent bandwidth. First, we compute $\overline{N}^*_{\text{call}}$ for the smallest satisfactory Ω_{call} with $B(\overline{\Omega}_{\text{call}}) = g_{\text{call}}$. Hence, similar to (7), we have:

$$\overline{N}^*_{\text{call}} = E^{-1}\left(g_{\text{call}}, \frac{\lambda_{\text{call}}}{\mu_{\text{call}}}\right). \tag{10}$$

The state $N_{\text{call}} = \overline{N}^*_{\text{call}}$ requires the largest burst layer equivalent bandwidth for maintaining g_{burst} and g_{cell}. Using (5), (9), (10), the call layer equivalent bandwidth is given by:

$$\hat{C}_{\text{call}}(N_{\text{path}} = \lambda_{\text{call}})$$
$$= \hat{C}_{\text{burst}}(\overline{N}^*_{\text{call}})$$
$$= \gamma\lambda_{\text{cell}}E^{-1}\left(g_{\text{burst}}, E^{-1}\left(g_{\text{call}}, \frac{\lambda_{\text{call}}}{\mu_{\text{call}}}\right)\frac{\lambda_{\text{burst}}}{\mu_{\text{burst}}}\right). \tag{11}$$

The bottom curve of Fig. 6 plots the equivalent bandwidth as a function of $\rho_{\text{call}} = \lambda_{\text{call}}/\mu_{\text{call}}$ for our numerical example given earlier. For simplicity, we assume a cell layer penalty $\gamma = 1$. The other three curves are plotted for the cases when both the burst layer and call layer GOS bounds are reduced by factors of 2, 4, and 10, respectively. These curves show that equivalent bandwidth is not very sensitive to changes in GOS requirements. The curves are concave, which show that traffic aggregation provides a substantial statistical advantage.

For heterogeneous and non-Poisson traffic, we have derived highly accurate and fast numerical algorithms for computing the equivalent bandwidth at the burst and call layers using transform methods [20]. Alternatively, we are also considering modification to (9), (11) using the equivalent random method [8, chap. 8] with matched first and second moments for the traffic process at each layer.

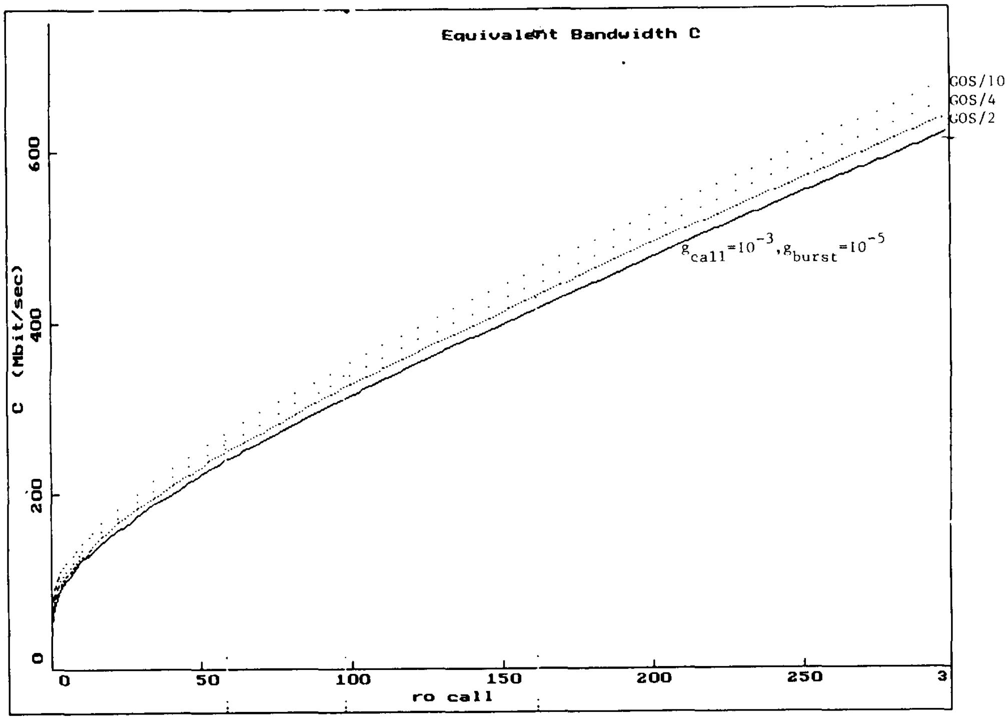


Fig. 6. Layered equivalent bandwidth for graphics service.

So far, we focus on the bandwidth required up to the call layer without regard to the actual path or route being chosen to accommodate the requirement. This function belongs to the route and path layers. The issue of dynamic routing for integrated broadband services is a complicated subject by itself and, hence, should be treated in a separate paper. In the ensuing sections, we shall ignore the route layer. The path layer is designed directly on top of the call layer, and a call for ω is considered blocked if the P_ω cannot admit the call. No alternate routing using a sequence of path groups is allowed. The path layer tries to provide the equivalent bandwidth $\hat{C}_{\text{call}}$ requested by the call layer for maintaining a bounded call blocking probability.

If an alternate routing layer is present between the path layer and the call layer, the following analysis remains valid. Blocking on the direct path group, instead of inducing a call blocking, induces the necessity for alternate routing. Hence, a larger probability of blocking in the direct path group can be tolerated because the alternate routing network substantially reduces the final blocking probability. Nevertheless, a properly configured path network should be engineered for a small (<0.1) probability of blocking for the direct path for several reasons. First, a small overflow probability from the direct path reduces the frequency of expensive call switching between two tandem paths for the alternate route. Second, too much alternate routing may cause system instability and subsequently inefficiency.

IV. Optimization for the Physical Path Layer

Using the notion of equivalent bandwidth at the call layer, we shall focus on the design and analysis of the path network layer. In this section, we formulate the mathematical program for designing the physical path network G_1 using a given facility network G_0. At the end of this section, we look at how a similar formulation can be used to design the facility network G_0, as well as the relationship between the path and facility network design problems. The optimization for the virtual path network is considered in the next two sections.

First, we are concerned with the size of the physical path for carrying traffic between an o–d pair ω. Given the GOS and traffic requirements, the equivalent bandwidth at the call layer is computed, which shall be represented by $\hat{C}_\omega$ henceforth instead of $\hat{C}_{\text{call}}$ for the ω of concern.

For a path p (which defines the end points ω), let f_p be the amount of bandwidth dedicated to the path. As defined before, let P_ω be the collection of p with the same end points ω, and Q_e be the collection of p using the same edge e.

A feasible physical path configuration must satisfy the following sets of constraints.

$$\sum_{p \in P_\omega} f_p \geq \hat{C}_\omega \qquad \text{for all } \omega \tag{C1}$$

$$u_e \triangleq \sum_{p \in Q_e} f_p \leq C_e \qquad \text{for all } e \tag{C2}$$

$$f_p \geq 0 \qquad \text{for all } p. \tag{C3}$$

The constraint set (C1) ensures that all path groups provide satisfactory GOS for the offered traffic. The constraint set (C2) ensures that the bandwidth assigned for each edge cannot exceed the capacity C_e. The constraint set (C3) imposes nonnegativity for assigning bandwidth.

Even though any set of f_p satisfying the constraint sets (C1, C2, C3) may suffice to meet the traffic and GOS requirement, we may choose a particular solution which limits the usage of the facility network. This choice is desirable because limiting usage may enable easier accommodation of additional traffic. Also, a feasible solution satisfying (C1, C2, C3) only may use an excessive number of paths between the same o-d pair. Hence, we may assume a linear cost for usage and, subsequently,

$$\text{Minimize} \sum_e \beta_e u_e \qquad \text{(P1)}$$

in which β_e is the cost per unit bandwidth used for the edge e, and the usage u_e is defined in (C2).

More generally, the linear function $\beta_e u_e$ can be replaced by the nonlinear function $\beta_e(u_e)$. The linear cost function is a reasonable cost assessment for most situations, particularly when a private network is built from leased lines. To avoid close to full utilization of the physical facility, a convex function β_e may be used. The minimization of (P1), a sum of convex (or linear) β_e, subject to (C1, C2, C3) always has a unique stationary optimal, and hence gradient search techniques can be efficiently applied. Also, the constraint set (C1) should be satisfied with equality for the minimum cost solution; otherwise, some f_p could be reduced for a lower cost.

In other situations, we may be concerned with minimizing a concave instead of a convex or linear $\beta_e(u_e)$. For example, a private network of leased lines embedded in a much bigger public network may enjoy a certain economy for leased capacity of a link. For this case and in the absence of a capacity constraint (C2), the optimization of a concave function subject to flow requirement (C1) gives a solution for which nonzero flow exists only on the shortest paths. However, the shortest path condition is necessary but not sufficient for proving global optimality.

Having formulated the physical path design problem, we now examine the associated facility network design problem. Let α_e be the facility cost per unit bandwidth. The objective is to choose C_e and the assignments f_p so as to:

$$\text{Minimize} \sum_e \alpha_e C_e \qquad \text{(P2)}$$

subject to the same sets of constraints (C1, C2, C3). (More generally, the constant α_e can be replaced by a nonlinear function $\alpha_e(C_e)$.) Hence, the facility network design problem is almost identical to the path network design problem, with β_e replaced by α_e and the flow constraint (C2) no longer capacitated since C_e becomes a variable.

If the cost functions α_e and β_e are identical, it seems that solving the facility network design problem would automatically give the solution for the path network design problem. However, the traffic pattern which determines $\hat{C}_\omega$ in the above problem may fluctuate from hour to hour. Giving the parameters f and $\hat{C}$ and the subscript t for each traffic period, the constraint sets (C1, C2, C3) become:

$$\sum_{p \in P_\omega} f_{p,t} \geq \hat{C}_{\omega,t} \qquad \text{for all } \omega, t \qquad \text{(C1')}$$

$$u_{e,t} \triangleq \sum_{p \in Q_e} f_{p,t} \leq C_e \qquad \text{for all } e, t \qquad \text{(C2')}$$

$$f_{p,t} \geq 0 \qquad \text{for all } p, t. \qquad \text{(C3')}$$

The facility network design problem is then the optimization of (P2) subject to (C1′, C2′, C3′). The problem becomes more difficult because the number of constraints is increased by a factor equal to the number of time periods.

The multiple period path network design problem is the optimization of (P1) subject to (C1′, C2′, C3′) for each time period t rather than for all t as in the facility network design problem. The path network design is necessary because the traffic pattern specified by $\hat{C}_{\omega,t}$ is poorly modeled during the solution of the facility network design problem.

Here, we can again apply the Principle of Layered Switching stated at the beginning of Section III. Due to the uncertainty of the traffic modeling, the facility network should be designed so as to bound the probability of blocking in path configurations. This blocking occurs when no feasible path configuration satisfies the constraint sets (C1, C2, C3). To evaluate the probability of blocking, an ensemble of traffic patterns may be specified. In practice, the evaluation may not be easy even if the ensemble is specified.

V. Optimization for the Virtual Path Network

The optimization of the physical path network is relatively simple because the bandwidth parameters for paths, namely f_p, are only weakly coupled for different ω through the constraint set (C2). This weak coupling results from effective traffic isolation among paths. On the other hand, effective isolation also makes sharing of link bandwidth among paths less flexible.

For virtual path networks, assignment of bandwidth to edges in G_1 (paths in G_0) is virtual. As shown in this section, virtual path networks allow more flexible sharing of the link capacity by paths but make the bandwidth assignment protocol and the GOS evaluation process much more complicated.

We describe a scheme in which each transmission link in the facility network is shared completely by the virtual paths using the link. For each o-d pair ω, we assume a set of candidate virtual paths P_ω. When a call arrives, one of the path $p \in P_\omega$ is chosen at random with probability r_p. Given an r_p for each virtual path in the network, we can compute the total offered traffic for the edge e. We assume that the call arrival rates to different paths for the same ω is divided according to r_p, whereas the call holding time as well as the burst layer statistics (burst arrival rate, burst bit rate, and burst duration) per completed call remain unchanged. Hence, we shall express the call layer equivalent bandwidth $\hat{C}_\omega$ as a function of λ_{call} only.

The amount of call traffic carried by the path p for the o–d pair ω is given by $r_p \lambda_{\omega(p)}$, in which $\omega(p)$ is the o–d pair at the two ends of the path p. The total offered traffic by all p for the link e is given by $\Sigma_{p \in Q_e} r_p \lambda_{\omega(p)}$. For each line e with capacity C_e, we engineer a usage level $u_e \leq C_e$ which should meet the GOS requirement for that link for the computed offered traffic. In other words, u_e must be more than the equivalent bandwidth for meeting the GOS requirement for the superposed traffic for all paths using e. Subsequently, we optimize (over the parameters r_p) the sum of u_e weighted by the usage cost. Expressed by a mathematical program, we have:

$$\text{Minimize } \sum_e \beta_e u_e \tag{P3}$$

subject to

$$\sum_{p \in P_\omega} r_p = 1 \qquad \text{for all } \omega \tag{D1}$$

$$\hat{C}_e \left(\sum_{p \in Q_e} r_p \lambda_{\omega(p)} \right) \leq u_e \leq C_e \qquad \text{for all } e \tag{D2}$$

$$r_p \geq 0 \qquad \text{for all } p. \tag{D3}$$

(P3) is identical to (P1) given earlier for optimizing the physical path network. (D1) maintains that the set of r_p is a probability distribution for P_ω. Its form is similar to (C1). (D2) maintains that the usage u_e must be sufficient to guarantee GOS, but must be within the capacity constraint C_e. Therefore, (D2) is analogous to (C2). Similar to (C3), (D3) imposes nonnegativity. Unfortunately, the constraint set (D2) is not convex in r_p, and hence local optimals may exist. In practice, the local optimals are either identical or close in their values.

The above mathematical program is relatively simple and tractable for configuring the virtual path network. Nevertheless, the inherent limitations and assumptions of the model require further explanations. First, we have the load sharing assumption which states that each call is offered to one path only according to the load sharing coefficients r_p. The use of other paths within a path group after a rejection by the first trial path is not allowed. In practice, using alternate paths within a path group may reduce call blocking. However, it is known that analyzing the performance of routing calls via alternate paths is very difficult. From computational experience, the above optimization tends to yield path groups containing a single path, namely that $r_p = 1$ for only one of the paths in P_ω. This results from the fact that blocking is substantially reduced by consolidating bandwidth in a single path rather than sharing load between two half-sized paths.

Second, (D2) guarantees GOS (bounds on call and burst blocking probabilities) per link rather than end-to-end for a call. These end-to-end blocking probabilities can be as high as the bound times the number of links in the path. Consequently, multiple link paths may suffer a poor GOS. To maintain good end-to-end GOS, we may reduce the probability bound by a factor equal to the maximum number of the links allowed in a path. We believe this more stringent probability bound may not affect the final solution significantly for good GOS. As shown in Fig. 6, the equivalent bandwidth is fairly insensitive to an overengineering multiplicative factor in the GOS for good GOS.

Third, traffic for an o–d pair offered to a path is actually offered simultaneously to all links in the path. Effectively, the offered traffic per link is reduced because of blocking occurring elsewhere on a path. For dynamic routing of telephony, this view leads to reduced offered load aproximations [22], [23] for calculating the blocking probability per link. Without a reduced offered load approximation, the constraint (D2) overestimates the equivalent bandwidth $\hat{C}$, giving a desirable overengineering effect.

A rather complicated mathematical program can be formulated to guarantee end-to-end GOS, using approximation methods to account for the second and third aspects mentioned above. However, we prefer the program (P3) for its simplicity and for the stated overengineering measures employed.

We would like to point out one major difference between the call setup procedures for physical versus virtual path methods. For the virtual path method, an arriving call is admitted on a virtual path if, for each link in the path, the admitted calls (including the call to be admitted) have satisfactory GOS (burst layer and lower). For the physical path method, call admission is simpler because the GOS condition is considered per path instead of per link. This simplicity is a consequence of traffic isolation among paths, and hence traffic need only be managed per path.

VI. A Simulation/Design Tool and Conclusion

A graphical simulation and design software tool called **CANeT** (computer-aided networking tool) has been developed using the object-oriented language Modsim operated on the UNIX platform. The tool is menu-driven and provides a graphical and interactive display of simulation and design results. The top-level menu allows the user to select the network layers described in this paper for study. The lower level menus allow the user to select methods (e.g., physical or virtual path methods) for each layer as well as options for graphical display.

Using the tool, we compare the bandwidth cost for the mathematical optimizations for physical path configurations (P1, subject to C1, 2, 3) versus virtual path configurations (P3, subject to D1, 2, 3). The facility network is the one shown in Fig. 1. The path network P_ω for each ω consists of all single or two hop paths between ω. If there is only one such path in P_ω [such as for $\omega = (a, c)$], we would add three hop paths to P_ω [such as the path (a, e, d, c)].

For all o–d pairs, we assume an offered traffic with parameters given for the video browsing service described in Section III. Traffic is offered from an origination to a destination in a unidirectional manner, with an equal amount of traffic going in the reverse direction. We started by assuming $C_e = 2$ Gb/s (capacity sum in both directions) for all links and solved the optimization problems for their optimal costs. We then decrease C_e gradually, consequently increasing the optimal cost (as the traffic between an o-d pair may be forced to use a more circuitous path, or split over multiple paths). Below a certain C_e, the optimization becomes infeasible.

For physical path configurations, we use a combination of the Frank-Wolfe and conjugate gradient methods to search for the optimal solution. The bandwidth constraint (C2) is incorporated into the objective function using a large quadratic penalty function. For virtual path configurations, we use a combination of the penalty function and gradient search methods. Convergence is achieved after 200-300 iterates for each configurations. Assuming 1 unit cost per Mb/s, we observe that the total cost is fairly insensitive to C_e above the value required for feasibility. For physical path configuration, this cost is roughly 6000 for $C_e \geq 1.35$ Gb/s, and for virtual path configuration, this cost is roughly 4000 for $C_e \geq 0.92$ Gb/s. Hence, there is a roughly 50% advantage of virtual path configuration over physical path configuration for this example. A similar advantage is seen comparing the C_e required for feasibility.

In conclusion, we have given a framework for layering connections from the path layer down to the ATM layer. An iterative graph theoretic procedure is used to generate alternatives and describe the layered bandwidth allocation process. The notion of equivalent bandwidth is used extensively for bandwidth allocation.

We have also formulated the path configuration problem for both the physical path network and the virtual path network. Preliminary evidence shows that good bandwidth advantage may be achieved using virtual path setup rather than physical path setup. However, this advantage is diminished if the individual path already has a significant degree of traffic aggregation.

Besides bandwidth efficiency, the issues of complexity and flexibility should also be considered for comparing virtual versus physical path networks. For broadband applications, the aggregated traffic on a link in the facility network may exceed hundreds of Gb/s. It remains to be shown that ATM cross connect can be implemented to serve nodes with a reasonable number of incident links. (ATM add-drop multiplexers may be used for nodes with a very small number of incident links.) On the other hand, facility cross connects, for which the switching fabric requires infrequent reconfiguration compared to the ATM cell time scale, can be implemented easily using conventional circuit switching methods. Also, the mathematical program for configuring the physical path network is simpler. Bandwidth checking for call setup is also easier for physical paths.

However, virtual path networks are potentially more flexible than physical path networks. First, path bandwidth is not required to be multiples of a predefined basic rate (say an OC-3 rate of 150 Mb/s). Second, bandwidth allocation is not tied directly to physical allocation (such as the actual assignment of timeslots within a TDM frame) and hence control can be exerted without necessarily acting at the physical layer.

Acknowledgment

The tool **CANeT** was developed by C.-f. Hsu, A. Kolarov, Y.-j. Li, J.-h. Li, and H.-s. Liou, who are supported in part by BellCore, NEC, and NSF.

References

[1] Special Issue on Asynchronous Transfer Mode, *Int. J. Digit. Analog Cabled Syst.*, vol. 1, no. 4, 1988.

[2] CCITT Recommendation I.212, "Broadband aspects of ISDN," CCITT IXth Plenary Assembly, Melbourne, Australia, 1988, vol. 3, fasc. III.7.

[3] CCITT Recommendation I.362,3, "B-ISDN ATM adaptation layer (AAL) functional description, specification," June 1990.

[4] Y. Okano, T. Kawata, and T. Miki, "Designing digital paths in transmission networks," in *Proc. GLOBECOM'86*, Houston, TX, 1986, pp. 25.2.1-25.2.5.

[5] S. Ohta, K. Sato, and I. Tokizawa, "A dynamically controllable ATM transport network based on the virtual path concept," in *Proc. GLOBECOM'88*, Ft. Lauderdale, FL, Nov 1988, pp. 39.2.1-39.2.5.

[6] R. G. Addie and R. E. Warfield, "Bandwidth switching and new network architecture," in *Proc. 12th Int. Teletraf. Cong.*, Turin, Italy, June 1988, pp. A.1.1-A.1.7.

[7] *IEEE J. Select. Areas Commun.*, vol. 5, Oct. 1987; vol. 6, Dec. 1988; vol. 9, Aug. 1991.

[8] J. Y. Hui, *Switching and Traffic Theory for Integrated Broadband Networks*. Boston, MA: Kluwer, 1990.

[9] —, "Resource allocation for broadband networks," *IEEE J. Select. Areas Commun.*, vol. 6, pp. 1598-1608, Dec. 1988.

[10] J. Filipiak, "M-architecture: A structural model of traffic management and control in broadband ISDNs," *IEEE Commun. Mag.*, vol. 27, pp. 25-31, May 1989.

[11] P. Boyer, "A congestion control for the ATM," presented at the 7th ITC Special. Sem. Broadband Technol.: Architect., Appl., Control and Perform., Morristown NJ, Oct. 1990, session 4.

[12] S. R. Amstutz, "Burst switching: An introduction," *IEEE Commun. Mag.*, Nov. 1983.

[13] G. R. Ash, R. H. Cardwell, and R. P. Murray, "Design and optimization of networks with dynamic routing," *Bell Syst. Tech. J.*, vol. 60, pp. 1787-1820, 1981.

[14] T. J. Ott and K. R. Krishnan, "State dependent routing of telephone traffic and the use of separable routing schemes," in *Proc. Int. Teletraf. Cong.*, 1985, vol. 11, pp. 5.1.A.5.1-5.1.A.6.

[15] R. J. Gibbens, F. P. Kelly, and P. B. Key, "Dynamic alternative routing-Modeling and behavior," in *Proc. Int. Teletraf. Cong.*, 1988, vol. 12 pp. 3.4A3.1-3.4A3.7.

[16] G. Woodruff and R. Kositpaiboon, "Multimedia traffic management principles for guaranteed ATM network performance," *IEEE J. Select. Areas Commun.*, vol. 8, Apr. 1990.

[17] Special Issue on Broadband Network Performance and Congestion Control, *Int. J. Digit. Analog Commun. Syst.*, Jan. 1990.

[18] J. Hui, "A congestion measure for call admission and traffic engineering for multi-layer multi-rate traffic," *Int. J. Digit. Analog Commun. Syst.*, vol. 3 pp 127-135, 1990.

[19] H. Ahmadi and R. Guerin, "Bandwidth allocation in high-speed networks based on the concept of equivalent capacity," presented at the 7th ITC Special. Sem. Broadband Technol.: Architect., Appl., Control and Perform., Morristown NJ, Oct. 1990, session 5.

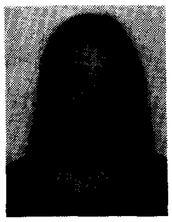
[20] J. Hui, "Layered required bandwidth for heterogeneous traffic," submitted to *IEEE INFOCOM'92*.

[21] M. J. Karol, M. G. Hluchyj, and S. P. Morgan, "Input versus output queueing on a space-division packet switch," *IEEE Trans. Commun.*, vol. 35, pp. 1347-1356, Dec. 1987.

[22] F. P. Kelly, "Blocking probabilities in large circuit-switched networks," *Adv. Appl. Prob.*, vol. 18, pp. 473-505, 1985.

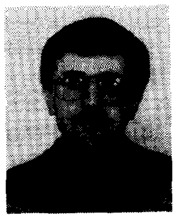
[23] S. P. Chung and K. W. Ross, "Reduced load approximations for multirate loss networks," to appear in *IEEE Trans. Commun.*
[24] J. Kaufman, "Blocking in shared resource environment," *IEEE Trans. Commun.*, vol. 29, Oct. 1981.

Joseph Y. Hui (S'82–M'83–SM'90), for a photograph and biography, see this issue, p. 1366.



Melike B. Gursoy received the B.Sc. and M.Sc. degrees in electrical engineering from Bogarici University, Istanbul, Turkey, and the Ph.D. degree in systems engineering from the University of Pennsylvania, Philadelphia.

Since graduating in 1988, she has been Assistant Professor in the Department of Industrial Engineering at Rutgers University, Piscataway, NJ. She worked as a Lecturer from 1980–1982 in the Department of Electrical Engineering, Middle East Technical University. Her research interests include control theory, stochastic processes, and their industry applications.



Nader Moayeri (S'79–M'86–SM'90) was born in Hamadan, Iran, on August 31, 1956. He studied electrical engineering at the Sharif (formerly Arya-Mehr) University of Technology, Tehran, Iran, from 1974 to 1978. He received the M.S.E.E., M.S.C.I.C.E., and Ph.D. degrees in electrical engineering systems from the University of Michigan, Ann Arbor, in 1980, 1981, and 1986, respectively.

He joined the Department of Electrical and Computer Engineering at Rutgers University, Piscataway, NJ, in September 1986, where he is currently an Assistant Professor. His research interests are in data compression, joint source and channel coding, mobile radio communications, information theory, and routing and flow control in data networks. He has served the Princeton section of IEEE in various capacities, among which was founding the Information Theory Chapter, where he serves as Chairman.

Roy D. Yates (M'91) received the B.S.E. degree from Princeton University, Princeton, NJ, in 1983, and the S.M. and Ph.D. degrees from the Massachusetts Institute of Technology, Cambridge, in 1986 and 1990, respectively, all in electrical engineering.

Since September 1990, he has been an Assistant Professor in the Department of Electrical and Computer Engineering at Rutgers University, Piscataway, NJ. His current research interests include queueing networks, high-speed networks, and cellular communication systems.