Routing Subject to Quality of Service Constraints in Integrated Communication Networks

With increasingly diverse QOS requirements, it is impractical to continue to rely on conventional routing paradigms that emphasize the search for an optimal path based on a predetermined metric, or a particular function of multiple metrics. Modern routing strategies must not only be adaptive to network changes but also offer considerable economy of scope.

Whay C. Lee, Michael G. Hluchyj, and Pierre A. Humblet

WHAY C. LEE is with the Networking Research Department at Motorola.

MICHAEL G. HLUCHYJ is Vice President and Chief Technology Officer at Summa Four. This work was performed while he was with Motorola.

PIERRE A. HUMBLET is with Eurecom Institute.

This paper was presented in part at INFOCOM '93, San Francisco, CA [1]

Traditional communication network architectures were designed to support users with homogeneous and simple quality of service (QOS) requirements. With increasing demand for a wide spectrum of network services, networking technologies that thrived on economy of scale are gradually giving way to new technologies that offer economy of scope. Modern communication networks must cater to users with diverse, fine-grain, and subjective QOS requirements. This task is accomplished by means of a wide spectrum of network control mechanisms operating over various time-scales. In a connection-oriented communication network, the transfer of information between two end users is accomplished by network functions that select and allocate network resources along an acceptable path. The logical association between the communicating end users is referred to as a call. The "chain" of network resources that support a call is a connection. Routing is a call-level network control mechanism through which a path is derived for establishing communication between a source and a destination in a network. This article proposes a call-level QOS framework in which QOS is specified in terms of QOS constraints, examines issues of routing subject to QOS constraints, and presents a call-by-call source routing scheme with rule-based fallbacks that depend on connection states.

Path selection within routing is typically formulated as a shortest path optimization problem, i.e., determine a series of network links connecting the source and destination such that a particular objective function is minimized. The objective function may be the number of hops, cost, delay, or some other metric that corresponds to a numeric sum of the individual link parameters along the selected path. Efficient algorithms for computing shortest paths have been used in communication networks (e.g., Dijkstra, Bellman-Ford) [2]. However, within the context of satisfying diverse QOS requirements, the computation becomes more difficult as constraints are introduced in the optimization problem. These constraints typically fall into two categories: link constraints and path constraints. A link constraint is a restriction on the use of links for path selection, e.g., available capacity on a link must be greater than or equal to that required by the call. Link constraints are relatively straightforward, as one simply removes links from the topology for shortest path computation that do not meet the link selection criteria. A path constraint is a bound on the combined value of a performance metric along a selected path (e.g., end-to-end delay through the network must not exceed what the call can tolerate). Path constraints make a routing problem intractable. In fact, a shortest path problem with even one path constraint is known to be NP-complete (see Shortest Weight-Constrained Path Problem in [3]).

The subjectivity of today's QOS requirements and the complex trade-off among them make it difficult to define an appropriate routing metric. Moreover, with distinct characteristics of different traffic types, the same metric is not universally applicable. The introduction of QOS negotiation further renders the meaning of an optimal path indeterminable. There is little or no additional utility for having a path whose associated QOS is more desirable than the user-specified QOS. On the other hand, there is considerable disutility for failing to find a path that meets the user-specified QOS. Hence, a new routing paradigm that emphasizes searching for an acceptable path satisfying various QOS requirements is needed for integrated communication networks.

Apart from issues of user-specified QOS requirements, there is also the problem of routing in a dynamic environment due to transient fluctuations in offered load, time-of-day changes in service demand, and incidental disruptions in the network (e.g., link failure and call preemption). In addition, the routing topology may also

0890-8044/95/$04.00 1995 © IEEE

change because of the possibility of dynamically adding and removing transmission facilities. Thus, modern routing strategies must be designed to adapt to changes in the network.

In this article, we consider the problem of routing in networks subject to QOS constraints. After providing an overview of prior routing work in the second section, we define various QOS constraints in the third section. In the fourth section, we present a call architecture that may be used for QOS matching. In the following section, we present a connection management mechanism for network resource allocation. Next, we discuss routing subject to QOS constraints. Following that we present fallback routing, and review some existing routing frameworks. In the following section, we present a new rule-based, call-by-call source routing strategy for integrated communication networks. A brief conclusion is provided in the final section.

Overview of Existing Routing Technologies

Many routing algorithms or strategies for connection-oriented and connectionless networks are found in the literature. We provide an overview below for a few that address issues of QOS and dynamic networks.

Alternate routing is widely used in telephone networks to provide dependable services. In [4], two fundamentally different classes of alternate routing algorithms, deterministic and randomized alternate routing algorithms, were investigated. Dynamic Non-Hierarchical Routing (DNHR) is a deterministic alternate routing scheme where the assignment of alternate paths is static, subject to time-of-day changes [5]. It was later extended to allow periodic update of the alternate assignment via a trunk status map [6]. DNHR eventually evolved into a more robust adaptive routing method in which a predetermined one-hop path is attempted first on a per-call basis, and if there is not enough capacity, all two-hop alternate paths are scanned to determine, if available, the one with the least load [7]. A similar approach known as Dynamic Traffic Management (DTM) makes routing decisions for groups of calls instead of on a call-by-call basis, and allows them to attempt a direct path and a recommended alternate path with the largest number of free circuits [8]. In addition to the previous techniques on which existing systems are based, a variety of other approaches have been suggested. In randomized alternate routing, each alternate path is selected with a given probability for a call. The probability assignment for each alternate path is continuously updated to reflect the likelihood of a successful attempt with the path. The more likely an alternate path permits a successful attempt, the higher is its probability of being selected. Early versions of stochastic routing are found in [9, 10]. In a recent version, known as Dynamic Alternate Routing (DAR), routing choices are reset whenever a call fails [11]. When a one-hop path is busy, the alternate two-hop path last successfully used is used once again for the next call overflow. If the alternate path is busy, calls are blocked and a new alternate path is selected at random. In fuzzy alternate routing, each alternate path is assigned a value between 0 and 1, determined by a "fuzzy membership" function, to indicate the relative order of selection of the path [12].

Conventional routing paradigms for data networks emphasize the search for an optimal path based on a predetermined metric, or a particular function of multiple metrics. In [13], an algorithm was proposed to solve a routing and flow control problem for an operating point with the optimal trade-off among delay, throughput, and fairness. In [14], a routing problem is formulated as a nonlinear combinatorial optimization problem, with the objective of maximizing a total traffic-dependent reward for the admitted calls subject to constraints on end-to-end cell loss probability and delay. In [15], a dynamic routing method that minimizes a long-run average cost of lost calls, using an optimization technique based on Markov decision theory, was proposed. Intra-domain routing protocols, such as the Open Shortest Path First (OSPF) protocol, support assignment of different type-of-service routing metrics based on different combinations of delay, throughput, and reliability [16].

Policy routing incorporates policy related constraints into path computation and packet forwarding functions for inter-domain communication [17]. It uses explicit policy advertisement along with topology information and a link-state style source routing protocol. Each administrative domain is governed by an autonomous administration, with distinct goals as to the class of customers it intends to serve, the QOS it intends to deliver, and the means for recovering its cost. The abstract policy route, a series of administrative domains, is specified by the end administrative domain as a form of source route, and each policy gateway selects the next actual policy gateway that is to be used to forward the packets.

Routing subject to multiple path constraints (e.g., cost and delay constraints) is a desirable feature in today's integrated networks in spite of its intractability. In [18], two heuristic approaches for solving a shortest path problem subject to multiple path constraints were proposed. In [19], an intractable multicast routing problem subject to a path constraint is heuristically reduced to constrained shortest path subproblems. It is common to formulate a routing problem subject to multiple path constraints as a multicriteria shortest path problem where each constrained path metric is taken to be a routing objective. A simple approach for multicriteria routing is to assume that the "optimal" path is a non-dominated path, one such that every other path has at least one path value greater than the corresponding path value of that path. With this assumption, one can generate the set of non-dominated paths so that the utility function can be applied to each non-dominated path to determine the desired path. A number of methods to generate these paths for a shortest path problem with two objective functions were reported in [20].

Call preemption has become increasingly useful for prioritized resource allocation in applications where there is a wide variety of traffic types. In [21], several versions of an optimal call preemption problem were shown to be computationally intractable, and some heuristic algorithms were proposed. Less efficient, but robust, preemption algorithms are found in [22]. In these algorithms, calls are normally routed using a table-driven routing algorithm, without considering their priority levels. When a high-priority call is blocked,

Routing subject to multiple path constraints (e.g., cost and delay constraints) is a desirable feature in today's integrated networks in spite of its intractability.

We propose a call-level QOS framework in which QOS is specified in terms of three classes of QOS constraints which may depend on the type of service of the call.

the call is routed, with preemption, along a path determined by means of flooding search messages all over the network. Some non-preemptive routing strategies for networks with priority classes are surveyed in [23]. These strategies resort to some means of either reserving resources for high-priority calls (e.g., separate networks, trunk reservation, trunk subgrouping), or extending the search for feasible paths (e.g., more choices for alternate paths, limited waiting for resources).

Work on rule-based routing is relatively scarce. In [24], Stach proposed an Expert Router that can monitor and predict a network's configuration to decide which path to use for each new call. By establishing a profile of the application and associating it with the speeds of the terminals, the Expert Router can quickly determine whether a call is delay-tolerant or not. Calls with high delay tolerance are assigned the "poorest acceptable paths," whereas calls with low delay tolerance are assigned the "best" available paths.

Optimization techniques using artificial neural networks have found their way into adaptive routing in communication networks [25]. A neural network architecture implemented in each node in a communication network is continuously trained to recognize the current status or topology of the communication network as long as there are messages passing through or received by the node. Despite its potential, many open issues remain to be resolved before the suitability of this approach could be adequately established for routing in today's integrated networks.

Quality of Service Constraints

QOS may be considered as a degree of conformance to user-specified service criteria. Many existing QOS frameworks either focus heavily on traffic management performance criteria (e.g., [26]), or accommodate fine-grain user-oriented quality requirements (e.g., [27]). The QOS framework in emerging ATM networks distinguishes QOS at different time scales [28]. It supports a set of individually specified QOS parameters (e.g., available cell rate, cell transfer delay, etc.), and permits them to be configured as standard profile sets [29]. Additional work on QOS framework can be found in [30-34]. An important element and challenge in the architecture of integrated networks is the ability to offer the diversity of QOS required by the different applications that use the network, and yet still make efficient use of network resources. In [35], one finds a survey of a large volume of research on QOS issues focusing on performance-oriented call admission control in high speed networks.

We propose a call-level QOS framework in which QOS is specified in terms of three classes of QOS constraints which may depend on the type of service of the call: performance constraints (e.g., throughput, delay), resource constraints (e.g., transmission medium, channel security), and priority constraints (e.g., establishment priority, retention priority) [36]. A performance constraint is a constraint on a directly perceivable measure of the quality of information transfer over a connection. A resource constraint is a restriction on the use of a given type of network resource with a particular set of characteristics. A priority constraint is a condition imposed on network resource allocation to provide different blocking probabilities to traffic of different priority classes.

Performance constraints may be negotiable or non-negotiable among the network and the end users. A negotiable constraint is specified in terms of a range of values bounded by a requested value and an acceptable value. A requested value is the most desirable performance level the user would like to have if resources are readily available. An acceptable value is the least desirable performance level the user would tolerate. A non-negotiable constraint is specified in terms of only an acceptable value. Although each performance constraint is basically a path-dependent QOS constraint, it can be implemented as a constraint on either a link attribute (e.g., throughput) or a path attribute (e.g., delay). A link attribute is a link parameter that is considered individually to determine whether a given link is acceptable for carrying a given connection, whereas a path attribute is an accumulation of an additive link parameter along a given path to determine whether the path is acceptable for carrying a given connection.

Resource constraints, which are subject to user definition, may be directly related to QOS (e.g., security), or indirectly related to QOS (e.g., carrier selection). Resources may refer to basic network elements (e.g., links), or aggregations of network elements (e.g., administrative domains). The effective topology used for path selection is determined by availability and acceptability of network resources with various resource attributes. Each resource attribute (e.g., transmission medium) is associated with a set of possible discrete attribute values (e.g., satellite, microwave). A resource constraint can be specified in terms of a subset of this set. A network resource is not acceptable for a call unless each of its resource attribute values belongs to the corresponding resource constraint set. The simplest resource constraints determine whether or not a given resource is acceptable for routing a call. They are predetermined and do not depend on the status of the network.

Priority constraints may be implemented in one of two approaches in a routing architecture: preemptive routing and non-preemptive routing. In a preemptive approach, network resources that have already been allocated to existing calls may be retrieved and used to accommodate new calls of greater importance. Thus, the blocking performance of high-priority calls is improved at the expense of disruption to low-priority calls. In a non-preemptive approach, calls of high priority are granted preferential access to network resources without bumping existing calls off the network. Thus, the blocking performance of high-priority calls is improved at the expense of the blocking performance of low-priority calls.

Call Architecture

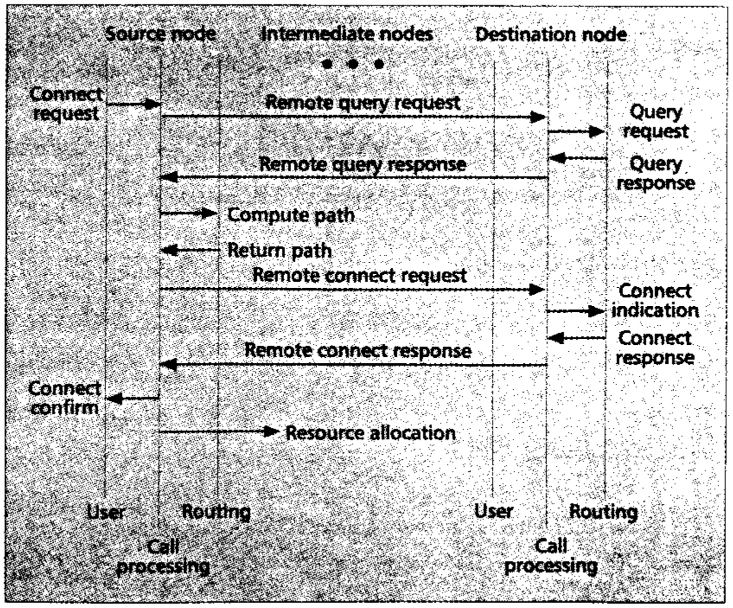
We now describe a call architecture that may be used to support QOS matching [37]. Call Processing and Routing are two key network control components residing in each node representing a switching system in a network. Together, they are used to provide each call a connection that satisfies all QOS constraints, and to maintain an acceptable level of QOS throughout the duration of the call. The flow of control signals between a pair of end users during a typical call setup is shown in Fig. 1.

When a source user initiates a connect request, Call Processing at the source node queries its peer entity at the destination node for destination-specific QOS information. Call Processing at the destination node in turn queries the destination user, and then returns the information obtained to Call Processing at the source node. The QOS constraints derived from the end users' QOS requirements are then consolidated into a consistent set of constraints that are assigned to the connection to be established between the end users. A set of QOS constraints is acceptable for call setup only if it is acceptable to both end users. QOS translation is carried out whenever it is needed in the QOS consolidation phase to account for various protocol processing overheads and to provide a bridge between the different interpretations of QOS on opposite sides of each protocol interface. Subsequently, Call Processing at the source node obtains from Routing an acceptable path that satisfies the consolidated QOS constraints. The QOS associated with this selected path is referred to as the available QOS. QOS negotiation is a procedure involving the network and the end users during call setup to determine the level of QOS that is agreeable by the end users and can be supported by the network. For each negotiable performance constraint, an agreed value is determined. During QOS negotiation, if it is determined that a path is available such that the network can provide a performance value that is better than the requested value, the path is accepted, but only the requested value would be guaranteed. If the network can only provide a performance value that is worse than the acceptable value, the call would be blocked, or it may have to preempt other calls.

When QOS negotiation is completed, a remote connect request is sent across the network and indicated to the destination user. The response is relayed back to Call Processing at the source node, which finally notifies the source user of the completion of QOS negotiation. Finally, Call Processing builds the network path by allocating resources from the source to the destination. In this phase, Call Processing derives from the agreed QOS meaningful network status information, and submits it to Routing for topology update. Call Processing also derives from the agreed QOS useful information for allocating network resources such that the agreed QOS can be guaranteed. Note that a call setup that involves QOS negotiation takes an amount of time that is of the order of two times the end-to-end round-trip delay. Compared to this delay, the time to compute a path for routing in a typical network is relatively small.

Connection Management

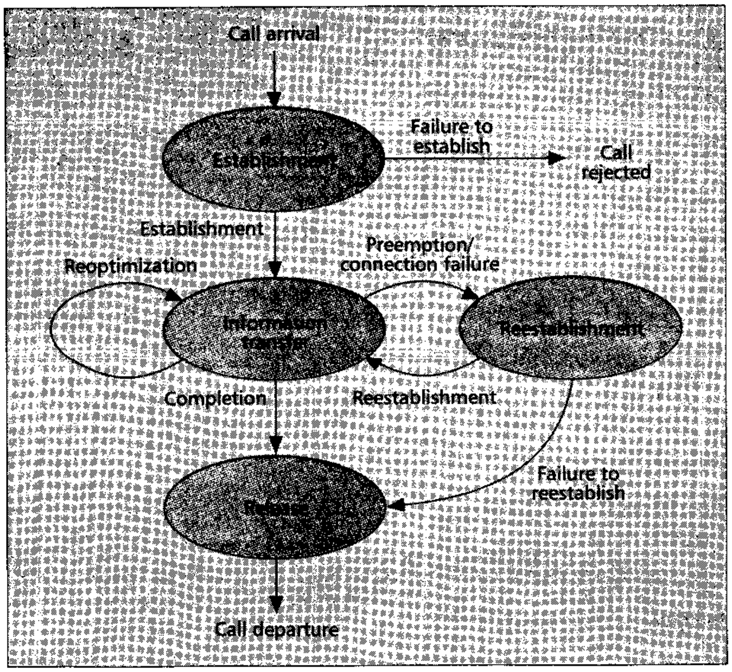
Connection management is a connection-level network control mechanism that is responsible for setting up, maintaining, and taking down connections. We show in Fig. 2 the transitions among four distinct connection states for connection management: Connection Establishment, Connection Reestablishment, Information Transfer, and Connection Release. Establishment refers to the setting up of a connection. Reestablishment is needed after an existing connection has been disrupted either by a network failure or by preemption. Reoptimization is performed by the network to conserve network resources utilized by established connections.



■ Figure 1. *Call architecture.*

When a call arrives at a network node, it enters the Establishment state. If there are not enough resources to support the call, the call is rejected. Otherwise, upon successful connection establishment (i.e., a path is available such that all QOS constraints are satisfied), the call enters the Information Transfer state. When a call in the Information Transfer state is completed, it enters the Release state, and the connection is subsequently taken down. When a call is in the Information Transfer state, its connection may fail or be preempted. Should this happen, the call enters the Reestablishment state, so that the network may automatically attempt to reestablish the connection by finding a new acceptable path. During reestablishment, the QOS constraints associated with an affected call are adjusted according to update rules that take into consideration the previously agreed value. Any missing or out-of-sequence protocol data units (e.g., cells in an ATM network) due to reestablishment could be taken care of by error recovery protocols at the link or transport layers. Provided an acceptable path is found, network resources are allocated along the new path from source to destination. Upon successful reestablishment, the call reenters the Information Transfer state. The length of time in which a connection reestablishment attempt may be repeated is limited by the connection reestablishment delay. Beyond this delay, the reestablishment procedure is aborted, and the call enters the Release state.

Reoptimization is carried out by the network, without direct user involvement, to find a more economical path or one that satisfies more stringent QOS constraints. It is useful for preventing connections from permanently using unnecessarily costly paths in the event that they happen to be established when the network is congested. Reoptimization can be triggered by the network admin-



■ Figure 2. *Connection states.*

istrator (periodically), or by time-of-day. The procedure for reoptimization is similar to that for reestablishment. During reoptimization for a call, the QOS constraints associated with the call are adjusted according to update rules such that they become more stringent, and the call remains in the Information Transfer state.

During call preemption, the preempting call must be in either the Connection Establishment state or the Connection Reestablishment state. During path computation, a call may only preempt lower priority calls that are in the Information Transfer state. However, for preemption due to resource contention while resources are being allocated to a new connection, the preempted call may be in any connection state.

Routing Subject to QOS Constraints

In this section, we examine issues of routing subject to performance constraints, resource constraints, and priority constraints.

Routing Subject to Performance Constraints

For each performance parameter, a path constraint is derived from its acceptable value. Unfortunately, a shortest path problem with path constraints is intractable [3]. The value of each constrained path metric may itself be a criterion for minimization to improve the chance of finding a path that satisfies the particular path constraint. However, a multicriteria shortest path problem is not a well defined optimization problem unless all the criteria are embedded in one utility function used as a single objective function for the shortest path problem.

There exists no efficient algorithm to solve the multicriteria shortest path problem with a general utility function. However, if a linear utility function (e.g., a weighted sum of delay and cost) is assumed, the problem reduces to a single-criterion shortest path problem with linearly combined link attribute values. One considerable drawback for using a linear combination of routing objectives is that the resulting routing performance is quite sensitive to the selected relative weights. For example, consider the routing problem shown in Fig. 3: Determine an acceptable path from source s to destination t. Let $P(m, n)$ denote the nth link on the mth path. For each link, the parameters C and D represent the respective cost and delay of the link. The first path is clearly the minimum cost path, and the second the minimum delay path. If the overall routing objective weighs cost and delay equally, then one can verify that the third path is indeed the optimal path. Should the relative weights be biased sufficiently one way or the other, either the first or the second path may be the optimal path. Note that it is not known beforehand that with equal weighting of cost and delay, the optimal path traverses many more hops than either the minimum cost path or the minimum delay path.

Routing Subject to Resource Constraints

To accommodate preferential selection of network resources, it is common practice to assign weights to links, and let link weights be incorporated in the routing objective function. The less desirable a resource attribute value of a link is, the heavier is the weight assigned to the link. This approach cannot consistently satisfy the user resource preference, since the minimization of the sum of link weights along a path does not guarantee that the weights of individual links are also minimized. Moreover, routing performance is also very sensitive to the weights assigned to the links. Consider the example shown in Fig. 4. In this example, the weight W for each link is indicated, with one exception. We let X be the unknown weight of the link. We examine how the minimum weight path depends on the value of X. For $0 \leq X < 4$, the first path is optimal, and it includes the link with weight X. For $X > 4$, the second path is optimal. Note that the second path traverses many more hops than the first path.

Resource attributes used to be specified with binary (include/exclude) choices when preferential resource selection is desirable. Today, many users demand multi-level preferential specification of resource attributes to support policy-oriented routing. For example, a user may specify each attribute value of a resource attribute in terms of one of four levels of resource preference: "required," "preferred," "don't_care," and "don't_use." A resource attribute is said to be preferentially dependent on another if its preference structure depends on that associated with the other resource attribute. For example, consider the following attribute sets: CONFIDENTIALITY with values "encrypted" and "unencrypted," and INTEGRITY with values "protected" and "unprotected." Suppose that a user permits either the combination of "unencrypted" and "protected," or the combination of "encrypted" and "unprotected." Then, no combination of confidentiality and integrity constraint sets can adequately represent this user's resource constraints. If all resource attributes are preferentially independent of one another, the user can specify preferences for one resource attribute at a time. For simplicity, it is desirable that all resource attributes are preferentially independent of one another.

Routing Subject to Priority Constraints

Preemptive routing algorithms that rely on flooding to determine desirable paths (e.g., [22]) are not very efficient in utilizing network resources. However, they may be acceptable for a military network where the QOS requirements for the high-priority calls are stringent and cannot be compromised. Non-preemptive routing algorithms that reserve network resources for high-priority calls are not very efficient either, since unclaimed reserved resources cannot be used by low-priority calls. Preemptive routing is a better approach, provided that there is an efficient means of path selection and a minimally disruptive connection establishment.

For the preemptive approach, flooding can be avoided if all the relevant information for routing is made available at each node via an efficient topology distribution protocol. Priority constraints can be specified with respect to connection states, and implemented as link constraints. Specifically, each call is assigned a priority number for the Connection Establishment state (establishment priority), the Connection Reestablishment state (reestablishment priority), and the Information Transfer state (retention priority). A preemption is permitted only when the priority of the preempting call is higher than the priority of the call to be preempted. The appropriate priority number used for comparison is the one that is associated with the connection state of the given call. The admissibility of the call on a given link thus depends on the appropriate priority level of the call. The priority numbers are also used to resolve resource contention when multiple calls are simultaneously trying to utilize the same network resources. In this case, the call with the highest appropriate priority is processed first.

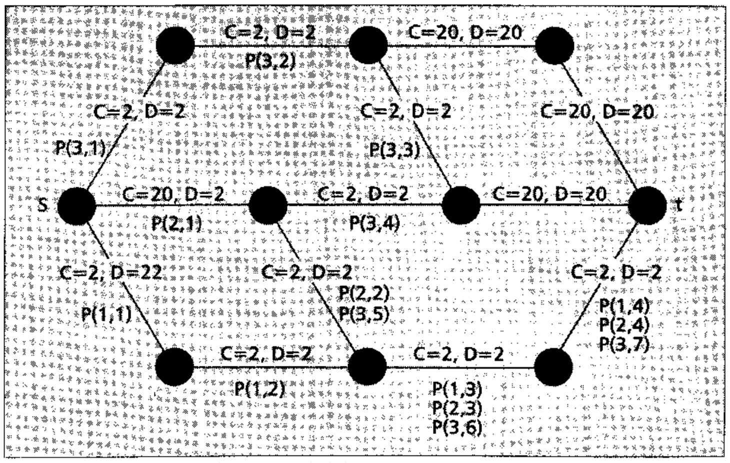
There is an undesirable cyclic effect, which occurs when a call with a high reestablishment priority preempts an existing call with a low retention priority, and is subsequently preempted by the latter because the retention priority of the former is lower than the reestablishment priority of the latter. As a result, the two calls may alternately switch between the Connection Reestablishment state and the Information Transfer state. The cyclic effect can be avoided if the retention priority of each call is required to be at least as high as its reestablishment priority.

Routing Framework

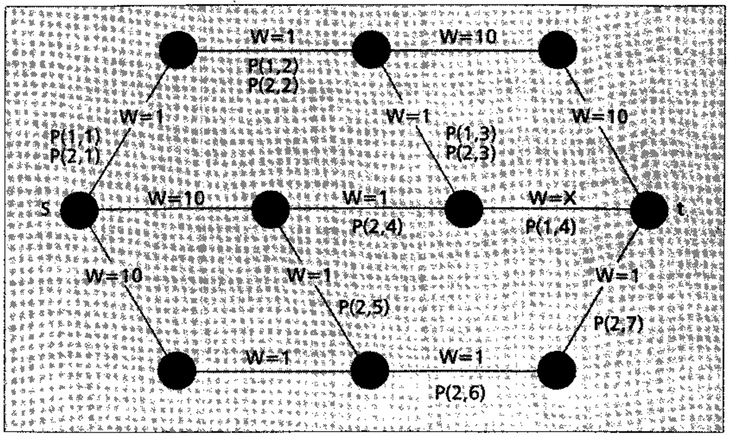
We now introduce fallback routing, review some existing routing frameworks, and compare their suitability for routing in integrated communication networks. Specifically, we consider five desirable aspects: economy of scale, economy of scope, fine granularity, path optimality, and adaptive capability. The comparisons are summarized in Table 1.

Call-by-Call Routing versus Table-Driven Routing

Call-by-call routing offers the flexibility of tailoring a path to the characteristics and QOS requirements of each call. Since path computation is done upon the arrival of each call, it permits fine-grain network control. There are increasing returns to scope because the computational overhead is independent of the number of call types. The approach also permits optimization of the path selected for each call. In today's integrated networks requiring call-level QOS matching, the time to compute a path is typically small compared to the round-trip delay of signaling to obtain the end users' QOS information for call setup. Provided that networks are well designed to support given call request rates, call-by-call routing offers great potential for QOS-sensitive routing.



■ Figure 3. *Routing with cost and delay metrics.*



■ Figure 4. *Link weight approach.*

In table-driven routing, all paths are precomputed and stored in routing tables. There are increasing returns to scale due to many calls being routed over the same path. However, this approach lacks fine-grain network control, and the selected paths cannot be optimal for all traffic types, unless a different table is used for each traffic type. The routing tables must adapt to topological changes.

Source Routing versus Hop-by-Hop Routing

In source routing, decisions for routing a call are made entirely at the source, based on global network configuration and status that are updated via a topology distribution protocol. Routing functions at intermediate nodes are thus relatively simple. With sufficient topology information available to the

Routing framework	Economy of scope	Economy of scale	Fine granularity	Path optimality	Adaptive capability
Call-by-call routing	x		x	x	x
Table-driven routing		x			
Source routing	x		x	x	
Hop-by-hop routing		x			x
Fallback rouitng	x		x	x	x
Alternate Routing		x			x

■ Table 1. *Routing framework comparison.*

source, it is possible to have fine-grain network control and optimization of desired paths. There are increasing returns to scope since the same topology information may be used for many traffic types. Source routing is hence very promising for path selection in modern integrated networks where there are increasingly many traffic types due to multimedia applications.

In hop-by-hop routing, routing decisions are distributed. Routing computation at intermediate nodes is non-trivial. With the use of routing tables, there are opportunities for economy of scale. However, routing tables require storage. In this approach, steps are needed to prevent loops. Although the approach can respond to failures quickly, the recovery is suboptimal because it retains the segment of the path from the source to the point of failure. It is difficult to use hop-by-hop routing to support call-level QOS matching (e.g., call preemption) because there is insufficient call specific information where routing decisions are made.

Fallback Routing versus Alternate Routing

In fallback routing, one sequentially computes paths based on a sequence of routing instances, until an acceptable one is available or the call will be blocked upon completion of a predetermined fallback sequence (Fig. 5). An instance of a constrained shortest path problem consists of link constraints, path constraints, and a routing objective function. Fallback routing adapts to changes in the network status, and permits alternate path computations to accommodate preferential resource constraints, call preemption, and other routing features that require multiple path computations per call setup. Fallback routing offers considerable economy of scope, for it accommodates heterogeneous users by computing alternate paths as they are needed. The sequence of fallback routing instances is either predetermined or selected in real time according to established rules. The routing instances can be specified to support fine granularity, provided pertinent QOS information is available.

In alternate routing, a set of predetermined paths stored in routing tables are attempted sequentially during each call setup for resource allocation until there is a successful attempt or the call will be blocked upon completion of the attempt sequence. The alternate paths may depend on traffic classes and the time-of-day. The use of routing tables offers economy of scale, but significant storage is needed. The routing tables may be periodically updated to adapt to existing network conditions. The manner in which alternate paths are attempted distinguishes among existing variations of alternate routing. Under light loads, alternate routing minimizes blocking probability. But, under heavy loads, blocking probability may be increased drastically as alternate paths used tend to consume more network resources.

Rule-Based Call-by-Call Source Routing

Using the terminology defined in the previous section, we propose a call-by-call source routing strategy with rule-based fallbacks for communication networks with integrated traffic subject to diverse QOS requirements. The strategy is to efficiently determine an acceptable path for each call given the current state of the network. As opposed to the traditional routing paradigm where the primary goal is to minimize the value of a routing objective function, the routing paradigm described below emphasizes meeting various routing constraints. Nonetheless, the shortest path algorithm is still used as a means to identify acceptable paths.

Routing with Rule-Based Fallbacks

The proposed rule-based routing strategy makes use of all available information to dynamically modify the fallback sequence of path computations according to network status and connection states. The strategy begins path computation with an initial routing instance that is determined by the connection state and the user QOS requirements. If no feasible path is found, stopping rules are used to decide whether a fallback computation is in order. In the fallback, a new routing instance with relaxed constraints is selected according to fallback rules. If a feasible path is found, the network will attempt to allocate network resources along the path for the call. This attempt may sometimes fail because of resource contention due to latency in topology updates. If this attempt fails, the call is blocked. A more sophisticated alternative to this approach is to allow crankback so that the source will select a new routing instance to fall back on, using the latest information derived from the unsuccessful attempt. Crankback routing is similar to alternate routing, except that alternate paths are not predetermined, but computed one at a time after each unsuccessful attempt to establishment a connection.

The proposed rule-based routing strategy follows a generic rule-based model for expert systems. The model consists of three modules: data base, knowledge base, and inference engine. The data base contains topology information that is updated via a topology distribution protocol. The knowledge base contains rules that are used to generate routing instances based on a predetermined routing policy and the QOS demanded by the users. Examples of these rules are described in the next three subsections. The inference engine is basically a sequential shortest path computer that also verifies path feasibility.

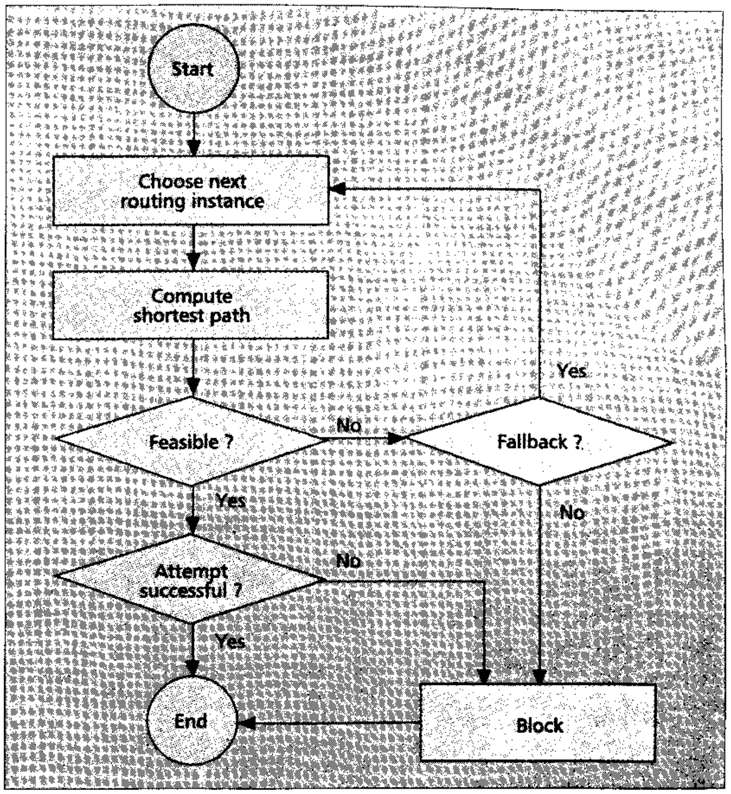
Fallback on Performance Constraints — Instead of computing an optimal path for the intractable constrained shortest path problem, the proposed rule-based routing strategy uses the following heuristic for fallbacks. The original routing problem without path constraints is first solved, and then the selected path is checked against the path constraints to determine feasibility. If the path constraints are not satisfied for the selected path, fallback allows the call to have one or more additional opportunities to search for a feasible path.

Instead of dealing with an unknown multi-dimensional utility function, the proposed rule-based routing strategy assumes that the routing criteria are ranked by the network user according to relative importance. The ordered criteria are then used to determine the shortest path objective functions for the initial and fallback path computations. For example, the initial objective function may be minimizing cost. Should the computed path fail to meet the specified delay constraint, a fallback objective function of minimizing delay could be used to find an acceptable path. This ordinal ranking approach is different from a cardinal ranking approach which relies on relative weightings.

Fallback on Resource Constraints — The proposed rule-based routing strategy can accommodate preferential resource constraints. User preference for resources can be translated into preferential sets of resource constraints. Each of these sets can be selected to define a routing instance according to some predetermined rules. Suppose that each attribute value of a resource attribute is specified in terms of one of the following levels of resource preference.

- REQUIRED: At most, one attribute value from a given attribute set may be configured "required." When an attribute value is configured so, only resources characterized by this attribute value may be used.
- PREFERRED: Resources characterized by attribute values configured "preferred" must be considered with priority over those characterized by attribute values configured otherwise, except for "required."
- DON'T_CARE: Resources characterized by attribute values configured "don't_care" may be considered, in addition to any configured "preferred," only when no acceptable path can be found otherwise.
- DON'T_USE: Resources characterized by attribute values configured "don't_use" must be avoided. At least one attribute value from a given attribute value set should be configured differently from "don't_use."

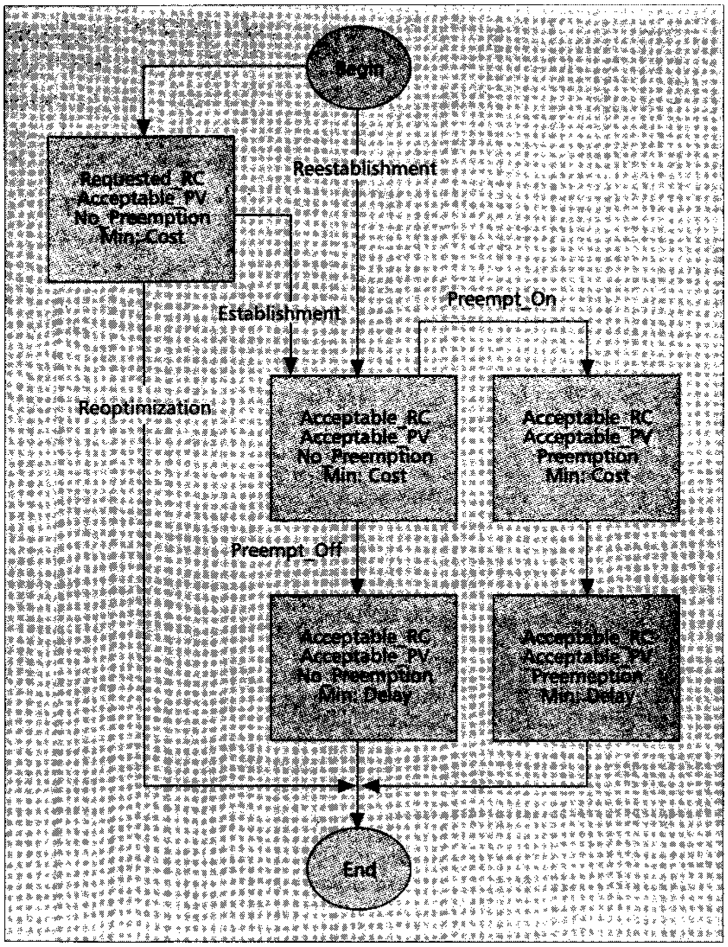
We translate the four levels of resource preference into two sets of resource constraints, namely "requested resource constraints," denoted Requested_RC, and "acceptable resource constraints," denoted Acceptable_RC. An attribute value may be included in one or both sets of constraints, and if it is not included in a set, it is considered excluded from it. For each resource attribute, the resource translation algorithm sequentially checks the numbers of attribute values specified "required," "preferred," and "don't_care." It detects an invalid configuration when there is more than one attribute value specified "required," or none specified other than "don't_use." If there is only one attribute value specified "required," it is included in both Requested_RC and Acceptable_RC. If no attribute value is specified "required," and at least one is specified "preferred," then those that are specified "preferred" are included in both Requested_RC and Acceptable_RC, whereas those specified "don't_care" are included only in Acceptable_RC. Else, if no attribute value is specified "required" or "preferred," and there is at least one attribute value specified "don't_care," those specified "don't_care" are included in both Requested_RC and Acceptable_RC.



■ Figure 5. *Routing with fallbacks.*

With fallback routing, paths are first computed using only resources satisfying the requested resource constraints, with a fallback to the less restrictive acceptable resource constraints.

Fallback on Priority Constraints — Preemptive routing for networks with priority classes can be disruptive, and the overall throughput is not increased because of call preemption. Nonetheless, since no resources are reserved for high-priority calls, low-priority calls are not likely to be blocked when the network is lightly loaded. Using source routing where topology information is made available to the source for efficient path selection, there is no need to flood search messages throughout the network to determine desirable paths. It is often possible to relax the resource constraints of a call so that its performance constraints can be met without unnecessary disruption to calls of lower priority. Since preemption is inherently disruptive, routing without preemption should be attempted first before preemption is considered. We refer to this approach as look-around-first preemption. The fallback approach to routing permits look-around-first preemption to avoid unnecessary preemption, by first attempting to find a feasible path without preempting any existing calls. If this should fail, then in searching for a feasible path, the fallback routing instance only accounts for the resources consumed by the same or higher priority calls.



■ Figure 6. *Rule-based routing.*

State-Dependent Fallback Rules

To facilitate call-level QOS matching, existing alternate routing strategies make use of simple rules for selecting predetermined alternate paths. Some intra-domain routing protocols, such as OSPF, rely on predetermined routing instances that depend on type of service. The proposed rule-based routing strategy makes use of a knowledge base that contains a variety of rules for selecting routing instances in real-time as they are needed for initial and fallback path computations. Its novel features include rules for selecting constrained routing instances, rules for falling back from one routing instance to another, and rules for determining when to fallback and when to stop.

A specific implementation of the proposed rule-based routing strategy is described below, and illustrated in Fig. 6. In this implementation, resource constraints are specified with preferences, as described earlier. In Fig. 6, PV stands for "performance value." The fallback scenario for each call depends on its connection state. In a normal situation for connection establishment without preemption, the initial routing instance is defined by the requested resource constraints, the path constraints, and the minimization of cost. If the initial routing attempt is unsuccessful, there will be a fallback on the resource constraints (i.e., the acceptable resource constraints will be used instead of the requested resource constraints). If the second attempt fails because the delay path constraint could not be satisfied, there will be a fallback on the routing objective function (i.e., delay instead of cost). If the third attempt is unsuccessful, the call will be blocked. If preemption is allowed, there will be a fallback on preemption before the fallback on the routing objective function. It is important to note that if the network is designed and dimensioned properly, routing fallbacks are rare except when the network is overloaded.

Reestablishment follows a similar fallback scenario, except that the first routing instance is skipped. For reestablishment, both the requested and acceptable values associated with each negotiable performance constraint are set to the previously agreed value, so that this value continues to be guaranteed. No fallback is allowed for reoptimization. A connection will not be reoptimized unless the fractional improvement in the path cost exceeds a predetermined threshold.

A connection may be interrupted when resources along its path degrade excessively in performance or are disabled. If preemption is allowed, a connection may also be bumped by another connection of greater importance. When a connection is interrupted, an attempt is made to reestablish the connection so that it resumes service quickly. Connection reestablishment must be completed within a predetermined interval, known as the connection reestablishment delay, or the connection must be released.

Conclusion

With increasingly diverse QOS requirements, it is impractical to continue to rely on conventional routing paradigms that emphasize the search for an optimal path based on a predetermined metric, or a particular function of multiple metrics. Modern routing strategies must not only be adaptive to network changes but also offer considerable economy of scope. To satisfy the need for future integrated networks (e.g., ATM) to accommodate traffic with diverse QOS requirements, we have proposed a call-by-call source routing strategy that makes use of rule-based fallbacks. This strategy provides a flexible platform on which routing can be efficiently carried out subject to performance constraints, resource constraints, and priority constraints.

Fallback routing, an integral part of the proposed strategy, is an iterative path calculation approach wherein routing constraints may be modified in each iteration based on the QOS requirements of the call, the connection state of the call, and dynamic network information. The outcome of a fallback path calculation is either a selected path or that there is no path that satisfies all the QOS constraints. Without fallbacks, a single-pass rule-based routing can neither accommodate preferential resource constraints nor look-around-first call preemption. Although alternate routing offers economy of scale (in the sense that the same computation performed can be used for many calls), it cannot efficiently support integrated traffic with diverse QOS requirements. The predetermined alternate paths impose unnecessary restriction on the search for a suitable path. Policy routing can accommodate to some extent various QOS constraints. However, the constraints must be translated into domain-based resource constraints, and there is no mechanism to deal with preferential resource constraints.

QOS-sensitive routing is an important feature in the emerging implementation of routing in ATM networks [38]. Here, the proposed routing architecture is hierarchical source routing with optional crankbacks. A variety of traffic-dependent QOS-related topology state parameters are advertised to support call-level QOS matching. Topology information at each hierarchical level is aggregated to trade-off fine-grain QOS matching for scaling in very large networks. Rule-based, call-by-call source routing is ideal for QOS-sensitive path selection within a routing domain where it is easy to distribute relatively accurate topology information. For inter-domain routing, where only partial topology information is available and different domains may use different intra-domain routing algorithms for path selection, it is difficult to guarantee efficient use of network resources. Nonetheless, fallback routing is still useful for preferential resource selection, prioritized multicriteria routing, look-around-first call preemption, and improving the chance of success for crankback and/or reroute after failure or call preemption.

Although alternate routing offers economy of scale, it cannot efficiently support integrated traffic with diverse QOS requirements.

Acknowledgment

The authors thank P. Kamat, D. Faulkner, and R. Constantin for many valuable comments and suggestions.

References

[1] W. Lee, M. Hluchyj, and P. Humblet, "Rule-Based Call-by-Call Source Routing for Integrated Communication Networks," *Proc. IEEE INFOCOM '93, 1993*, pp. 987-993.

[2] D. Bertsekas and R. Gallager, Data Networks, (Prentice-Hall, 1987).

[3] M. R. Garey and D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, (W. H. Freeman, 1979).

[4] D. Mitra and J. B. Seery, "Comparative Evaluations of Randomized and Dynamic Routing Strategies for Circuit-Switched Networks," *IEEE Trans. on Commun.*, vol. 39, no. 1, Jan. 1991, pp. 102-116.

[5] G. R. Ash, R. H. Cardwell, and R. P. Murray, "Design and Optimization of Networks with Dynamic Routing," Bell Labs. Tech. Journal, vol. 60, no. 8, Oct. 1981.

[6] G. R. Ash, "Use of a Trunk Status Map for Real-Time DNHR," *Proc. of the 11th Int'l Teletraffic Congress*, Kyoto, Japan, Sept. 1985.

[7] G. R. Ash and B. D. Huang, "An Analytical Model for Adaptive Routing Networks," *IEEE Trans. on Communications*, vol. 41, no. 11, Nov. 1993, pp. 1748-1759.

[8] J. Regnier and W. H. Cameron, "State-Dependent Dynamic Traffic Management for Telephone Networks," *IEEE Commun. Mag.*, vol. 28, no. 10, Oct. 1990, pp. 42-53.

[9] K. S. Narendra, E. A. Wright, and L. G. Mason, "Application of Learning Automata to Telephone Traffic Routing and Control," *IEEE Trans. on Systems*, Man, and Cybernetics, vol. SMC-7, no. 11, Nov. 1977, pp. 785-792.

[10] V. V. Marbukh, "Investigation of a Fully Connected Channel Switching Network with Many Nodes and Alternative Routes," Simulation of Behavior and Intelligence, (Plenum Publishing Corporation, 1984), pp. 1601-1608.

[11] P. B. Key and G. A. Cope, "Distributed Dynamic Routing Schemes," *IEEE Commun. Mag.*, Oct. 1990, pp. 54-64.

[12] A. Krasniewski, "Fuzzy Automata as Adaptive Algorithms for Telephone Traffic Routing," *Proc. IEEE ICC '84*, pp. 61-66, May 1984.

[13] S. Chang, "Fair Integration of Routing and Flow Control in Communication Networks," *IEEE Trans. on Commun.*, vol. 40, no. 4, April 1992, pp. 821-834.

[14] F. Lin and J. R. Yee, "A Real-Time Distributed Routing and Admission Control Algorithm for ATM Networks," *Proc. IEEE INFOCOM '93*, 1993, pp. 792-801.

[15] A. Kolarov and J. Hui, "Least Cost Routing in Multiple-Service Networks," *Proc. IEEE INFOCOM '94*, 1994, pp. 1482-1489.

[16] J. Moy, "OSPF Version 2," RFC 1247, July 1991.

[17] D. D. Clark, "Policy Routing in Internetworks," *Internetworking: Research and Experience*, vol. 1, 1990, pp. 35-52.

[18] J. M. Jaffe, "Algorithms for Finding Paths with Multiple Constraints," *Networks*, vol. 14, 1984, pp. 95-116.

[19] V. P. Kompella, J. C. Pasquale, and G. C. Polyzos, "Multicast Routing for Multimedia Communication," *IEEE/ACM Trans. on Networking*, vol. 1, no. 3, June 1993.

[20] M. I. Henig, "The Shortest Path Problem with Two Objective Functions," *European Journal of Operational Research*, 25, 1985, pp. 281-291.

[21] J. A. Garay and I. S. Gopal, "Call Preemption in Communication Networks," *Proc. IEEE INFOCOM '92*, 1992, pp. 1043-1050.

[22] R. P. Lippmann, "New Routing and Preemption Algorithms for Circuit-Switched Mixed-Media Networks," *Proc. IEEE MILCOM '85*, 1985, pp. 660-666.

[23] G. Gopal, A. Kumar, and A. Weinrib, "Routing in a Circuit-Switched Network with Priority Classes," *Proc. IEEE INFOCOM '89*, April 1989, pp. 792-799.

[24] J. F. Stach, "Expert Systems Find a New Place in Data Networks," in Networking Software, Ungaro, ed., Data Communication Book Series, (McGraw Hill, 1987), pp. 75-83.

[25] C. Wang and P. N. Weissler, "The Use of Artificial Neural Networks for Optimal Message Routing," *IEEE Network Mag.*, March/April 1995, pp. 16-24.

[26] ANSI X3.102-1983, "American National Standard for Information Systems — Data Communications Systems and Services — User-Oriented Performance Parameters."

[27] J. S. Richters and C. A. Dvorak, "A Framework for Defining the Quality of Communications Services," *IEEE Commun. Mag.*, vol. 26, no. 10, Oct. 1988.

[28] H. Gilbert, O. Aboul-Magd, and V. Phung, "Developing a Cohesive Traffic Management Strategy for ATM Networks," *IEEE Commun. Mag.*, vol. 29, no. 10, Oct. 1991, pp. 36-45.

[29] ATM Forum Traffic Management Draft Specification, Version 4.0, April 1995.

[30] A. Campbell, G. Coulson, and D. Hutchison, "A Quality of Service Architecture," *ACM SIGCOMM Computer Communication Review*, vol. 24, no. 2, April 1994, pp. 6-27.

[31] A. A. Lazar, A. Temple, and R. Gidron, "An Architecture for Integrated Networks that Guarantees Quality of Service," *Int'l J. Digital and Analog Comm. Systems*, vol. 3, no. 2, 1990.

[32] A. A. Lazar and G. Pacifici, "Control of Resources in Broadband Networks with Quality of Service Guarantees," *IEEE Commun. Mag.*, vol. 29, no. 10, Oct. 1991, pp. 66-73.

[33] J. Jung and A. Gravey, "QoS Management and Performance Monitoring in ATM Networks," *Proc. IEEE GLOBECOM '93*, 1993, pp. 708-712.

[34] J. Jung and D. Seret, "Translation of QoS Parameters into ATM Performance Parameters in B-ISDN," *Proc. IEEE INFOCOM '93*, 1993, pp. 748-755.

[35] J. Kurose, "Open Issues and Challenges in Providing Quality of Service Guarantees in High-Speed Networks," *ACM SIGCOMM Computer Commun. Review*, vol. 23, no. 1, Jan. 1993, pp. 6-15.

[36] W. Lee and P. Kamat, "Integrated Packet Networks with Quality of Service Constraints," *Proc. IEEE GLOBECOM '91*, Dec. 2-5, 1991, pp. 8A.3.1- 8A.3.5.

[37] W. Lee and P. Kamat, "Quality of Service Matching for Integrated Fast Packet Networks," *Proc. IEEE GLOBECOM '92*, Dec. 6-9, 1992, pp. 931-937.

[38] ATM Forum PNNI Draft Specification, April 1995.

Biographies

WHAY C. LEE [M '89] received a B.S. in economics, B.S., M.S., E.E., and Ph.D. degrees in electrical engineering from MIT. In 1982, he was an intern at COMSAT Laboratories. In November 1988, he joined the Networking Research Department of Motorola Codex, Mansfield, Massachusetts, where he conducted applied research in routing, connection management, and quality of service issues in high speed integrated cell relay networks. Since 1994, he has been representing Motorola in the PNNI Subworking Group of the ATM Forum.

MICHAEL G. HLUCHYJ [F '94] received a B.S.E.E. from the University of Massachusetts-Amherst, and S.M., E.E., and Ph.D. degrees in electrical engineering from the Massachusetts Institute of Technology. In 1981, he joined the technical staff at AT&T Bell Laboratories, and in 1987, he became director of Networking Research at Motorola Codex, where he led efforts in the architecture, design, and analysis of all levels of traffic management for high-speed, integrated cell relay networks. In 1994, he became vice president and chief technology officer at Summa Four, Manchester, New Hampshire, where he is currently responsible for identifying and assessing strategic market requirements, technology trends and architectural directions for Summa Four's future products.

PIERRE A. HUMBLET [F '93] received his electrical engineering degree from the University of Louvain, Belgium, in 1973, and M.S.E.E. and Ph.D. degrees from MIT, where he became a professor of electrical engineering in 1978. In 1993, he joined the Eurecom Institute, France. His teaching and research interests are in the area of communication systems, particularly digital mobile networks and optical networks. He is a consultant with a number of companies, most recently with Motorola and IBM.