Source-Oriented Topology Aggregation with Multiple QoS Parameters in Hierarchical ATM Networks

Turgay Korkmaz and Marwan Krunz
Department of Electrical & Computer Engineering
University of Arizona
Tucson, AZ 85721
{turgay,krunz}@ece.arizona.edu

Abstract—**In this paper, we investigate the problem of topology aggregation (TA) for scalable, QoS-based routing in ATM networks. TA is the process of summarizing the topological information of a subset of network elements. This summary is flooded throughout the network, and is used by various nodes to determine appropriate routes for connection requests. A key issue in the design of a TA scheme is the appropriate balance between compaction and the corresponding routing performance. The contributions of this paper are twofold. First, we introduce a source-oriented approach to TA, which provides better performance than existing approaches. The intuition behind this approach is that the advertised topology-state information is used by *source* nodes to determine tentative routes for connection requests. Accordingly, only relevant information to source nodes needs to be advertised. We integrate the source-oriented approach into three new TA schemes that provide different tradeoffs between compaction and accuracy. Second, we extend our source-oriented approach to multi-QoS-based TA. A key issue here is the determination of appropriate values for the multiple QoS parameters that are associated with a logical link. Two new approaches to computing these values are introduced. Extensive simulations are used to evaluate the performance of our proposed schemes.**

keywords: QoS-based routing, topology aggregation, PNNI, scalable routing, ATM networks.

I. INTRODUCTION

Wide-scale deployment of ATM networks necessitates the provisioning of scalable routing mechanisms that take into account the quality-of-service (QoS) requirements of prospective connections as well as the available network resources. The Private Network-to-Network Interface (PNNI) protocol [10] of the ATM Forum provides a scalable, hierarchical framework for routing connection requests across large ATM networks. This framework is based on three fundamental routing techniques. First, it is a link-state (or, topology-state) routing protocol, in which each node in the network acquires knowledge about the entire network from the topological information that is flooded by other nodes. Second, it is a source routing protocol, in which the originating node of a connection request determines a tentative end-to-end route for this request using its knowledge of the entire network. The tentative route must have a high probability of passing the admission test at intermediate switching nodes. Otherwise, the request will be rejected or rerouted, resulting in longer connection establishment times and lower call throughput. Third, PNNI uses hierarchical routing approach in which nodal and link-state information is summarized at multiple levels in the hierarchy before being flooded throughout the network. This process is known as topology aggregation (**TA**), as it involves the mapping of a collection of interconnected nodes (*peer group* nodes) into a more compact, standardized representation. While TA is needed to ensure the scalability of the routing mechanism, the reliance on aggregate information in determining an appropriate route for a connection request may result in an *infeasible* route, which would ultimately fail the call admission test at some intermediate node. Therefore, an efficient TA scheme must provide an adequate balance between topology compaction (less advertised information) and "lossyness" (impact of compaction on routing performance). In certain cases, TA is also needed for security reasons to hide the details of the underlying subnetwork.

In PNNI, TA consists of nodal and link aggregation. Nodal aggregation refers to the process of summarizing a peer group (PG) into a more compact representation that comprises a "logical node" at the next higher level of the hierarchy. Link aggregation refers to the process of representing a set of parallel links between two peer groups by a single logical link. The two types of TA are described in Figure 1. Here, the network consists of ten physical nodes (e.g., ATM switches), which are structured into a two-level hierarchy. At Level 1 the ten nodes are clustered into two peer groups, which are individually aggregated and represented as two logical group nodes (LGN) at the higher level. The process of mapping a PG into a LGN is known as nodal aggregation It is performed by a designated node in each PG, known as the peer group leader (PGL). In Figure 1 there are three links that connect the two PGs at Level 1. When represented at Level 2, these links are collapsed into one logical link; a process known as link aggregation.

After aggregating the topological information of its own PG, a PGL maps this aggregate topology into a standardized versatile representation known as the complex-node. In its simplest (and default) form, the complex-node consists of a simple node that is characterized by one value per QoS parameter (e.g., the diameter). This default representation is conveyed in the form of a symmetric 'star' topology that consists of a nucleus and several *ports*. Typically, these ports are the border nodes of the child PG of the LGN. Such default representation may be too lossy and could seriously degrade the routing performance, particularly when the weights associated with various links in the PG differ significantly. Alternatively, PNNI allows for a more detailed representation

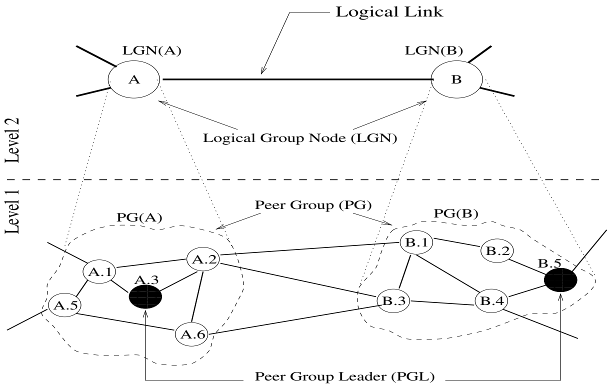


Fig. 1. Example of topology aggregation in PNNI.

in which the links between the nucleus and the ports (the *spokes*) can have different weights. In PNNI's terminology, these asymmetric links are called *exceptions*. PNNI also allows the complex-node representation to include direct links (or *bypasses*) between the ports of a LGN.

Related Work

While PNNI defines a hierarchical structure and a flexible representation mechanism for performing TA, it does not specify any TA schemes. Such schemes are left for vendor differentiation. Accordingly, several TA schemes have been proposed in the literature (see [15]). In many of these schemes, TA is performed in two steps. In the first step, a fully connected mesh of the border nodes of a PG is constructed, with a direct link between each pair of border nodes. It can be argued that for a single QoS parameter, this step is "lossless" in the sense that it retains all the distances between the border nodes of the original graph (as explained later, this losslessness does *not* hold under multiple QoS parameters). The second step involves mapping the full mesh into a more compact topology, such as symmetric-node (simple-node), star [15], minimum spanning tree [14], and t-spanner [2]. Graph reduction is performed by pruning several links of the full mesh. The compact topology is then represented as a complex-node, which is broadcasted to the rest of the network.

One problem in the above approach is that the amount of lossyness that results from graph reduction is not known in advance, and can vary depending on the actual values of the QoS parameters. To remedy this issue, some researchers proposed new TA approaches that minimize the average distortion (i.e., lossyness) in a least-square sense [7], [5]. In [12], [16] the authors present several heuristic algorithms for route selection in the presence of inaccurate topological information, including inaccuracies that are caused by TA. The effects of several TA schemes on routing performance have been studied by simulation [4], [13]. There have been several other proposals for hierarchical routing in literature [19], [18], [1], [6]. In [11] the authors present a new method called link-vector algorithm (LVA) for distributed routing in IP networks. LVA ensures that each router receives the topological information that it needs. Recently, LVA has been extended to area-based link-vector algorithm (ALVA) for hierarchical routing in the Internet [6]. In ALVA, only relevant information is advertised to the rest of the network. A similar philosophy lies behind our work in this paper (note that ALVA is intended for distributed routing in IP networks).

Contributions and Organization of the Paper

The main contributions of this paper are twofold. First, we introduce a novel, source-oriented approach to TA, in which only the relevant topological information of a PG is advertised. Relevance is defined relative to the *source* nodes that compute the tentative routes for connection requests. Since the relevant information about a PG varies from one source node to another, our source-oriented approach involves advertising different compact topologies of the same PG throughout different border nodes. This is in sharp contrast to conventional approaches in which the same topological information is advertised to nodes outside a PG. Based on our approach, we present three source-oriented TA schemes, which provide different tradeoffs between compaction and lossyness. We show that our schemes achieve better performance than their conventional counterparts. Our second contribution is the application of the source-oriented TA approach to a hierarchical network with multiple QoS parameters. More specifically, we propose two new schemes for obtaining the multiple QoS values of a logical link. Such schemes are used in the first step to TA; namely, the construction of a full-mesh of border nodes. Extensive simulations are used to evaluate the performance of our schemes and contrast them with other existing schemes.

The rest of the paper is organized as follows. In Section II, we describe the inefficiency of the standard approach to topology dissemination in PNNI. In Section II-B we present a generic source-oriented TA methodology. Based on this methodology, three different source-oriented TA schemes are presented in Section III. Two strategies for obtaining the weights of a logical link under multiple QoS parameters are given in Section IV. In Section V we evaluate the performance of our schemes and contrast them with conventional TA schemes. Finally, the paper is concluded in Section VI.

II. TOPOLOGY AGGREGATION

In this section, we first discuss the conventional TA approach and point to its inefficiency. Then, we introduce our source-oriented TA approach. For illustration purposes, we consider a simple, two-level hierarchical topology in which ten nodes (i.e., ATM switches) are clustered into five PGs (Figure 2). We focus on the aggregation of the central PG, which consists of six nodes. Links are bidirectional and asymmetric, i.e., both directions have different QoS values.

A. Conventional Approach

In a conventional TA scheme, the first step to aggregate the central PG is to construct a full-mesh of its border nodes, as shown in Figure 3(a) (logical links are indicated by dashed lines). Since this representation may lead to excessive advertisements, it is further reduced to a more compact topology

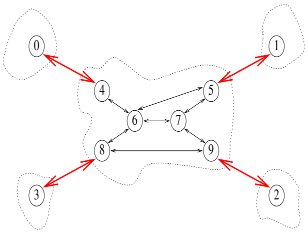


Fig. 2. Two-level hierarchical network topology.

such as a 'star' (Figure 3(b)). After performing these steps, the PGL of the central PG maps the 'star' topology into a complex-node representation, and advertises it to neighboring PGs. Thus, Nodes 0, 1, 2 and 3 will receive the same topological information regarding the central PG. Of course, this information may be incomplete, causing some degradation in the routing performance. Note that the PGL advertises the same information about its PG to outside nodes irrespective of the relevance of this information to the route selection mechanism at these nodes.

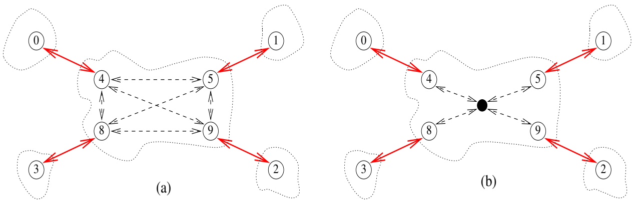
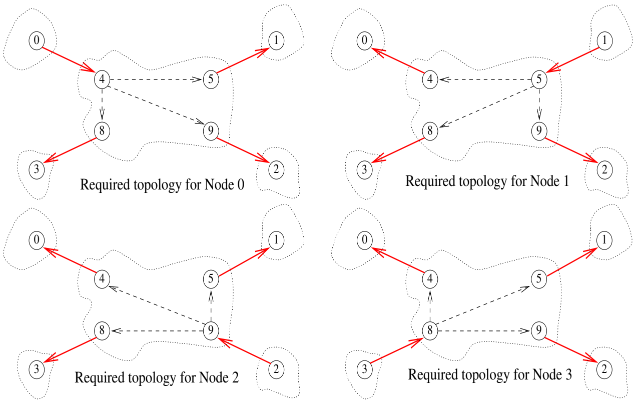


Fig. 3. TA of the central PG (a) full-mesh generation; (b) graph reduction.

B. Source-Oriented Approach

It can be noted from Figure 3(a) that Nodes 0, 1, 2, and 3 do not need to receive information about all the logical links of the central PG. Consider, for example, Node 0. Starting from this node, suppose that the only way to reach the central PG is through Node 4, i.e., Node 0 has one entry to the central PG. If Node 0 wants to compute a route for a connection request that traverses the central PG, it only needs to know the logical links that originate from Node 4 and end at some other border node, namely, links $4 \rightarrow 5$, $4 \rightarrow 8$, and $4 \rightarrow 9$ ($i \rightarrow j$ refers to a logical link from Node i to Node j). Other information about the central PG are redundant from the standpoint of Node 0. Similarly, Nodes 1, 2, and 3 need different partitions of the full-mesh representation of the central PG, as illustrated in Figure 4.

Note that a logical link typically represents the best path between two border nodes in a PG, so it incorporates all the information that other PGs need to know about routing a request from one of these border nodes to the other border nodes. Advertising different partitions of a full-mesh representation to different neighbors based on their needs results in better compaction than the full-mesh advertisement with the same routing performance. We refer to this type of aggregation as *source-oriented* TA. It is obvious that this approach accommodates asymmetric links and multiple QoS parameters as long as the weights of the logical links are appropriately computed.



Required topology for Node 0

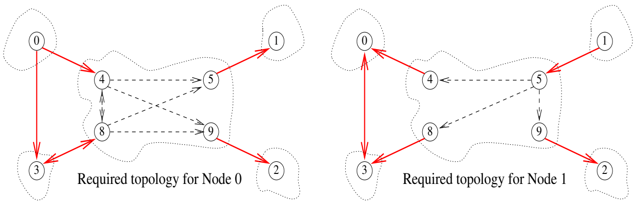
Required topology for Node 1

Required topology for Node 2

Required topology for Node 3

Fig. 4. Required topology information for Nodes 0, 1, 2, and 3.

In the previous example, the inter-PG connectivity is sparse (a tree structure), resulting in $\mathcal{O}(M)$ advertisements, where M is the number of border nodes of the central PG. However, if an outside node can reach a given PG through k different border nodes, then this node needs to receive k different partitions of that PG. For example, if in Figure 2 a link exists between Nodes 0 and 3, then Node 0 needs to know two different partitions of the central PG, while Node 1 needs only one partition, as shown in Figure 5. In other words, different outside nodes require different amounts of information about the same PG depending on their connectivity to this PG. Therefore, the overhead for advertising and storing the topological information varies from one node to another. On average, this overhead is $\mathcal{O}(kM)$, where M here is the average number of border nodes of a PG and k is the average number of entries to PG from an outside node, $1 \leq k \leq M$. The complexity of the source-oriented TA approach is between $\mathcal{O}(M)$ and $\mathcal{O}(M^2)$, depending on the network topology. For dense topologies, one needs to further reduce the amount of advertised information at the expense of some lossyness.



Required topology for Node 0

Required topology for Node 1

Fig. 5. Required topology views for Nodes 0 and 1.

III. SOURCE-ORIENTED TA SCHEMES

In this section, we present three source-oriented TA schemes. The first one is more appropriate for sparse topologies, and is a promising alternative to the conventional advertisement of a full-mesh. The other two schemes are aimed at dense topologies. First, let's define a new topology called *quasi-star*, which is a star-like topology whose nucleus is a border node of a PG and whose leaves are the remaining $M-1$ border nodes. The logical links connecting the nucleus to the leaves are extracted from the full-mesh representation of the underlying PG. Four such quasi-stars were shown earlier in Figure 4. A quasi-star can be easily encoded in the PNNI complex-node format using $M-1$ bypasses.

Scheme 1: Unified quasi-stars

In this scheme, after constructing a full-mesh of border nodes, the PGL of a given PG partitions this full-mesh into M quasi-stars. The PGL associates each quasi-star with one border node and maps it into a complex-node representation with different $M-1$ bypasses. The M representations are then advertised to the outside through various border nodes. An outside node may receive different quasi-stars of the same PG. It stores the first received quasi-star in its database, and forwards it to all of its neighbors except the one from which the advertisement was received. If the node receives a subsequent advertisement of the same quasi-star, then it checks the sequence number (SN) in this advertisement. If this number is smaller than or equal to the SN of the currently stored advertisement, the node discards the newly arriving advertisement. Otherwise, if the SN of the new advertisement is larger than the one in the database, the new advertisement is stored in place of the older one and is forwarded to all neighbors as above. Since advertisements are sent periodically, the larger the SN the more updated the corresponding advertisement. So far, the processing is standard. If the node receives the advertisement containing a different quasi-star of the same PG, the node combines this quasi-star with the ones in the database, and forwards only this new quasi-star to all of its neighbors except the one from which the advertisement was received. Different quasi-stars of the same PG can be easily recognized by checking the identity of their nuclei and one of their bypasses.

Since a faraway node receives all the partitions it needs, this TA scheme has the same accuracy as advertising the full-mesh, but often with less advertisement overhead. Another important advantage of advertising different quasi-stars is as follows. Suppose that the state information of the PG has changed, causing significant changes to only one quasi-star. Then, the PGL can advertise only this quasi-star (i.e., compact triggered updates). In contrast, the conventional approach would have to advertise the complete full-mesh. The advertisement overhead in Scheme 1 may still be an issue, and it depends on the density of the inter-PG connectivity, as discussed earlier. For dense topologies, further reduction in this overhead is desirable. For this purpose, we provide two other TA schemes.

Observation

In making routing decisions, source nodes often prefer shorter paths over longer ones. As a matter of fact, it has been shown that restricting routing to short paths achieves efficient resource utilization in QoS-based routing [17], [13]. Thus, providing a node with detailed topological information on the shortest paths and more compact information on longer ones should intuitively give a good balance between compaction and routing performance. Consider a node with multiple entries to a faraway PG. Under our source-oriented approach, this node will receive different advertisements of the same faraway PG. The first received advertisement is expected to traverse one of the shortest paths from the faraway PG to this node. The reverse path is also expected to be one of the shortest paths since propagation delay is symmetric and it is the dominant delay component in high-speed networks [8]. The receiving node acquires more accurate information on the short paths by storing and forwarding only the topological information in the first received advertisement. Based on this observation, we propose the following two schemes.

Scheme 2: Source-oriented simple-node

In the conventional simple-node scheme, a PG is aggregated into a single node with one value for each QoS parameter. This value, which is typically the diameter of the underlying graph [15], is advertised to all neighbors of the PG. We now present a source-oriented version of the simple-node scheme. Suppose the PG consists of l border nodes, $n_1, n_2, \ldots, n_l$. Let d_{ij} be the distance from n_i to n_j, for all i and j. The distance is the cost of the best path between two nodes. The cost of a path depends on the particular QoS parameter. So, for example, with respect to the delay measure, the cost of a path is the sum of link weights along that path. In contrast, for the bandwidth measure, the cost of a path is given by the minimum weight of a link along that path (i.e., the bandwidth of the bottleneck link). The best path is one with maximum or minimum cost, depending on the QoS parameter (e.g., minimum delay, maximum bandwidth, etc.). To aggregate a PG, the PGL first constructs a full-mesh of border nodes. The weight of each logical link in this full-mesh corresponds to the distance between two border nodes. From the full-mesh, the PGL selects the cost of the *worst* path from a given node n_i to every other node. For example, for the delay parameter the PGL computes

$$d_{max}(i) \triangleq \max_{j \neq i} d_{ij} \text{ for } i = 1, \ldots, l$$

The l different values $d_{max}(i)$, $i = 1, 2 \ldots, l$, are individually advertised to the outside through the corresponding border nodes. These l advertisements carry the same SN. When an outside node receives an advertisement from a given PG for the first time, it stores this advertisement in its database and forwards it to all of its neighbors except the one from which the advertisement was received. Subsequent advertisements that carry the same or smaller SN are discarded. While advertisements with the same SN possibly represent other perspectives ($d_{max}(i)$ values) of the same PG, they will not be considered because they have traversed longer paths than the first advertisement. The overhead per advertisement is still $\mathcal{O}(1)$ as in the conventional simple-node scheme.

Scheme 3: Source-oriented star

One common TA approach is based on the asymmetric star topology [15]. Here, the full-mesh of M border nodes is reduced into a star whose leaves are the border nodes.

The center of the star is a fictitious nucleus. In contrast to the default (symmetric) representation, the links between the nucleus and the leaves have different weights, which can be determined in various ways [13], [4], [7]. In the following scheme we will consider the average-case approach. To aggregate a PG, the PGL first constructs a full-mesh of border nodes. The weight of each logical link in the full-mesh is the distance between two border nodes. To aggregate the full-mesh into a star, the PGL computes the weights of links from each border node to the fictitious nucleus, and vice versa. In our case, the weight of a link from a border node n_i to the fictitious nucleus f is is taken as the average weight of logical links that originate from this border node in the full-mesh. More formally, from each node n_i,

$$d_{if} = \frac{1}{l-1} \sum_{j=1, j \neq i}^{l} d_{ij} \quad \text{for } i = 1, 2, \ldots, l$$

The weight of a link from the fictitious nucleus f to a border node n_j is the average weight of logical links that end at this border node. More formally, for each node n_j, the PGL computes

$$d_{fj} = \frac{1}{l-1} \sum_{i=1, i \neq j}^{l} d_{ij} \quad \text{for } j = 1, 2, \ldots, l$$

The complexity of the conventional asymmetric star approach is $\mathcal{O}(M)$, which is a compromise between a full-mesh and a simple node. Nevertheless, the asymmetric star approach is still lossy. If the lossyness is unacceptable, additional links (exceptions and bypasses) can be added to represent "significantly different or important" topology information [10], [15]. PNNI guidelines recommend that the total number of advertised values per QoS parameter is less than $3M$. Here, we propose a source-oriented asymmetric star approach that requires advertising $3M - 1$ values per QoS parameter (so the overhead is still $\mathcal{O}(M)$). This approach relies on the observation presented above; the most important information is provided in the first received quasi-star.

The source-oriented star scheme first finds the star representation of the full-mesh as described above. Then, it combines each quasi-star with the same star and constructs M distinct complex-node representations of the same PG. Each representation consists of the same star but a different quasi-star. When an outside node receives a complex-node representation of a PG for the first time, it stores it and forwards it to its neighbors except the one from which the advertisement received. If this outside node receives another complex-node representation with the same or older SN from the same PG, the outside node discards it. An example of this scheme is shown in Figure 6 which depicts Nodes 0 and 1 views of the central PG.

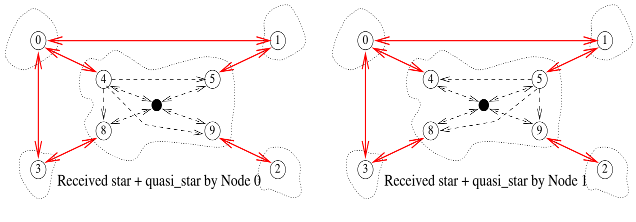


Fig. 6. TA of the central PG from the view point of Nodes 0 and 1.

IV. AGGREGATION UNDER MULTIPLE QoS PARAMETERS

One fundamental step in TA, in general, is the assignment of weights (or QoS values) to logical links. This step is used to construct a full-mesh of border nodes, with a logical link connecting each pair of border nodes (in each direction). The full-mesh is then followed by graph reduction, as explained before. Essentially, a logical link is an aggregation of all the paths between two border nodes in the original topology. For a single QoS parameter, the weight of a logical link is simply the cost of the "best" path between the underlying two border nodes. For additive parameters (or *metrics*), the cost of a path is given by the sum of link weights along that path. Cell delay is an example of an additive parameter. For non-additive parameters (or *attributes*), the cost of a path is given by either the minimum or the maximum link weight along that path. Bandwidth is an example of an attribute in which the cost of the path is given by the minimum link weight. The best path is one with the maximum or minimum cost, depending on the nature of the QoS parameter (e.g., for bandwidth maximum is best, for delay minimum is best).

Determining appropriate weights for logical links under multiple QoS parameters is not so simple. In particular, a best path with respect to one parameter is not necessarily the best one with respect to another. Hence, a strategy is needed to assign appropriate weights for logical links. In this section, we first illustrate the problem of weight assignment under multiple QoS parameters. We then summarize the current strategies to computing the weights of logical links. Finally, we propose two new enhanced strategies.

A. Logical Links with Multiple Parameters

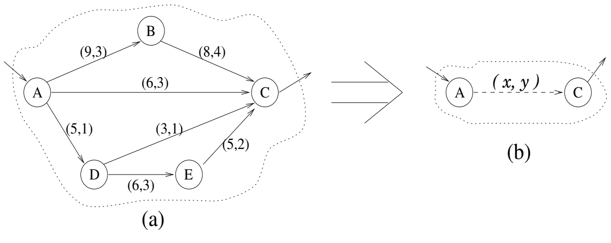


Fig. 7. A PG in which each link has two parameters (BW, delay).

Figure 7(a) shows a PG in which each physical link is associated with two QoS parameters: bandwidth (BW) and delay. There are four distinct paths from Node A to Node C: ABC, AC, ADC, and ADEC, with corresponding bandwidth costs 8, 6, 3, and 5, and with associated delays 7, 3, 2, and 6, respectively. We would like to represent these four paths by a single logical link with appropriate bandwidth and delay values (x, y). Note that ABC is the best path with respect to bandwidth, while ADC is the best one with respect to delay. The (BW,delay) parameter space is shown in Figure 8. In this figure, the costs of the four paths are shown in cir-

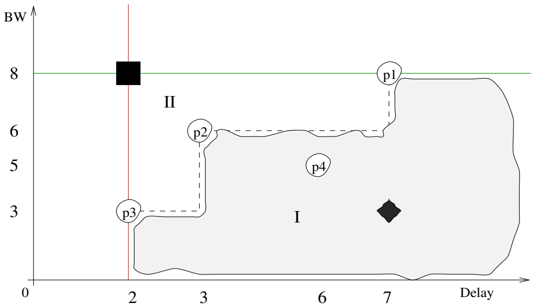


Fig. 8. Partitioning the QoS-parameter space based on physical paths.

cles. The combination of the best values with respect to BW and delay is represented by a black box, while the combination of the worst values is represented by a black diamond. The shaded area (Region I) represents the actual admissible region, i.e., a connection request whose QoS requirements falls in this region will be accepted. Otherwise, the request will be rejected because there is no physical path that meets its QoS requirements. Since the four physical paths will be represented by a single logical link (a point in the parameter space), the exact boundaries of the admissible region are not known to an outside node that needs to perform route selection. However, the weights of the logical link should approximate as much as possible the actual admissible region.

B. Conventional Approaches

Existing approaches can be classified as follows:

- **Single-Path-Parameters Approach** (SPPA): The SPPA finds the best path according to a single QoS parameter, and then assigns all the QoS parameters of this path to the logical link [15]. The problem is how to choose this single most-important parameter. Assuming that such a parameter has been chosen and that the logical link has been assigned the QoS values of one of the physical paths, then the admissible region seen by an outside node is restricted to a portion of the actual admissible region. For example, if the minimum-delay path is used to represent the logical link from Node A to Node C (p3 in Figure 8), then the admissible region as seen by an outside node is restricted to the area that lies to the bottom right of p3. SPPA achieves zero crankback rate at the expense of poor network utilization since some requests will be unnecessarily rejected at the originating node.
- **Multiple-Path-Parameters-Best-Case Approach** (MPPBCA): MPPBCA assigns the best QoS values to the logical link (i.e., the black box in Figure 8). It is an aggressive approach since it is quite possible that none of the physical paths can simultaneously support all the advertised QoS values [15]. From the standpoint of a node performing route selection, the perceived admissible region consists of Regions I and II. Of course, only requests that fall in Region I will eventually be admitted, so there is some probability of crankback.
- **Multiple-Path-Parameters-Worst-Case Approach** (MPPWCA): This approach finds the worst value for each QoS parameter from different paths (i.e., the diamond in Figure 8), and assigns these values to the logical link. In this approach, the utilization is expected to be relatively low because many requests that can be supported by a path from A to C will be unnecessarily blocked or routed through different nodes. However, this approach is less sensitive to the dynamic changes in the network state than the previous two approaches.

C. Two New Strategies

We now introduce two new strategies for assigning the weights of a logical link under multiple QoS parameters.

Strategy 1: Closest-Single-Path Approach (CSPA)

Let $P = \{p_1(Q_1^1, ..., Q_1^K), ..., p_n(Q_n^1, ..., Q_n^K)\}$ be the set of n physical paths between two border nodes, where each path p_i is associated with K QoS parameters $Q_i^1, Q_i^2, \ldots, Q_i^K$. Suppose that the K parameters consist of L attributes (non-additive parameters), which are followed by $K-L$ metrics (additive parameters). Without loss of generality, we assume that for attributes the higher the value the better the path (i.e., best is maximum), while for metrics the smaller the value the better the path (i.e., best is minimum). Let

$$Best_{Q^k} = \max\{Q_i^k \mid 1 \le i \le n\} \text{ for } k = 1, ..., L$$

be the best values of the first L QoS parameters (the attributes), and

$$Best_{Q^k} = \min\{Q_i^k \mid 1 \le i \le n\} \text{ for } k = L+1, ..., K$$

be the best values of the last $K-L$ QoS parameters (the metrics). For the ith path $p_i(Q_i^1, Q_i^2, \ldots, Q_i^K)$, we define its *stretch factor* s_factor_i as

$$s\_factor_i \triangleq \left(\sum_{k=1}^{L} \frac{Best_{Q^k}}{Q_i^k} + \sum_{k=L+1}^{M} \frac{Q_i^k}{Best_{Q^k}} \right) \qquad (1)$$

Note that each term in the above equation is greater than or equal to one, thus $s\_factor_i \ge K$ for all i. For the example in Figure 7, the set of paths is given by $P = \{p_1(8,7), p_2(6,3), p_3(3,2), p_4(5,6)\}$. The stretch factor for each path $p_i(BW_i, D_i) \in P$, $i = 1,2,3,4$, is given by

$$s\_factor_i = \frac{Best_{BW}}{BW_i} + \frac{D_i}{Best_D} \qquad (2)$$

According to CSPA, the weights of the logical link are taken from the QoS values of the physical path that has the minimum stretch factor. In the example in Figure 7, $p_2(6,3)$ has the minimum stretch factor, so its QoS values are assigned to the logical link.

Strategy 2: Modified-MPPBCA

The conventional MPPBCA finds the best QoS value for each parameter and assigns it to the logical link. Using its information about this logical link, the path selection algorithm at a faraway source node assumes that connection requests that fall in the Regions I and II in Figure 8 can be routed from Node A to Node C. Thus, crankback can possibly occur with probability that corresponds to the probability that the requested QoS requirements belongs to Region II. To reduce the probability of crankback while maintaining a high level of utilization, we propose a method which excludes part of Region II. For this purpose, we add one parameter to the set of best QoS values that are associated with a logical link. The additional parameter is $s\_factor \triangleq \min\{s\_factor_i \mid 1 \leq i \leq n\}$. Accordingly, the logical link is assigned $K+1$ values $(Best_{Q^1}, Best_{Q^2}, \ldots, Best_{Q^K}, s\_factor)$. Consider now a connection request with QoS requirements $(Req_{Q^1}, Req_{Q^2}, \ldots, Req_{Q^K})$. As before, the K parameters consist of L attributes and $K-L$ metrics, in this order. A faraway source node decides that there is a high probability that the logical link can support the requested connection if the following conditions are *simultaneously* satisfied:

$$\begin{aligned} Req_{Q^i} &\leq Best_{Q^i}, \ \forall i = 1, 2, \ldots, L \qquad (3)\\ Req_{Q^i} &\geq Best_{Q^i}, \ \forall i = L+1, L+2, \ldots, K \qquad (4)\\ s\_factor &\leq \left(\sum_{i=1}^{L} \frac{Best_{Q^i}}{Req_{Q^i}} + \sum_{i=L+1}^{K} \frac{Req_{Q^i}}{Best_{Q^i}} \right) \qquad (5) \end{aligned}$$

Otherwise, the logical link is not considered in route selection. In the special case of two QoS parameters (bandwidth and delay), the above conditions reduce to

$$(Req_{BW} \leq Best_{BW}) \text{ and} (Req_D \geq Best_D) \text{ and} \\ \left(\left(\frac{Best_{BW}}{Req_{BW}} + \frac{Req_D}{Best_D} \right) \geq s\_factor \right)$$

where Req_{BW} and Req_D are the bandwidth and delay requirements of the connection request, respectively. Figure 9 depicts the resulting tentative admissible region under Modified-MPPBCA for the previous example. In addition to Region I, the tentative admissible region also includes a portion of Region II.

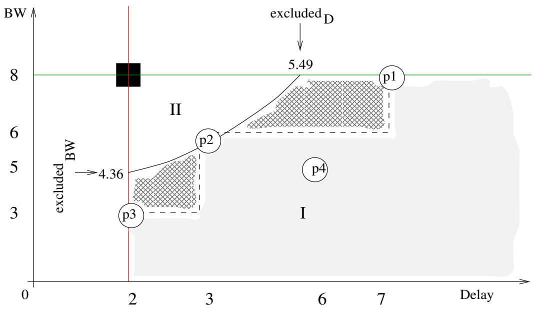


Fig. 9. Admissible region based on Modified-MPPBCA.

The remaining important question is that: Given a path p which consists of l logical and physical links and given a connection request with end-to-end QoS requirements, how can a routing algorithm decide whether this path is likely to meet the end-to-end QoS requirements? Assume each link i has $K+1$ QoS values $(Q_i^1, Q_i^2, \ldots, Q_i^K, s\_factor_i)$. Since the best and the actual QoS values of a physical link are the same, the s_factor of a physical link is K (number of QoS parameters). Consider now a connection request with QoS requirements $(Req_{Q^1}, Req_{Q^2}, \ldots, Req_{Q^K})$. As before, the K parameters consist of L attributes and $K-L$ metrics, in this order. Without loss of generality, we assume that for attributes the total cost of a path is the minimum weight of a link along that path, while for metrics it is the sum of link weights along the path. Define

$$total_{Q^k} = \begin{cases} \min\{Q_i^k \mid 1 \leq i \leq l\} & \text{for } k = 1, \ldots, L \\ \sum_{i=1}^{l} Q_i^k & \text{for } k = L+1, \ldots, K \end{cases}$$

Note that these total weights are found based on the best QoS parameters, but which may not be associated with the same physical path. For each QoS parameter along the path, we can individually calculate the best weight after excluding part of Region II. Let

$$excluded_{Q^k} = \min\{ \frac{Q_i^k}{(s\_factor_i - (K-1))} \mid 1 \leq i \leq l\} \qquad (6)$$

be the best weight for $k = 1, \ldots, L$ (the first L QoS parameters) after excluding part of Region II, and

$$excluded_{Q^k} = \sum_{i=1}^{l} Q_i^K (s\_factor_i - (K-1)) \qquad (7)$$

be the best weight for $k = L+1, \ldots, K$ (the last $K-L$ QoS parameters). For example, Figure 9 shows these weights for bandwidth ($excluded_{BW}$) and delay ($excluded_D$). Finally, $total_{s\_factor}$ for the entire path can be estimated aggressively based on the minimum excluded QoS parameter as follow:

$$total_{s\_factor} = \min \left\{ \frac{excluded_{Q^k}}{total_{Q^k}}, \frac{total_{Q^{k'}}}{excluded_{Q^{k'}}} \right\} + (K-1) \qquad (8)$$

where $1 \leq k \leq L$ and $L+1 \leq k' \leq K$. In the routing decision, if the following conditions

$$\begin{aligned} Req_{Q^i} &\leq total_{Q^i}, \ \forall i = 1, 2, \ldots, L \qquad (9)\\ Req_{Q^i} &\geq total_{Q^i}, \ \forall i = L+1, L+2, \ldots, K (10)\\ total_{s\_factor} &\leq \left(\sum_{i=1}^{L} \frac{total_{Q^i}}{Req_{Q^k}} + \sum_{i=L+1}^{K} \frac{Req_{Q^i}}{total_{Q^i}} \right) \qquad (11) \end{aligned}$$

are *simultaneously* satisfied, then the path p is likely to meet the end-to-end QoS requirements of the connection request. Otherwise, the path will not meet the QoS requirements of the request.

V. SIMULATION RESULTS

In this section, we present simulation results for TA based on a single and multiple QoS parameters. Using these results, we compare various TA schemes presented in this paper. In addition to TA, there are other factors that affect the routing performance such as network topology, traffic load, call holding times, database update policies and intervals, and routing algorithms [17], [3]. However, to enable an accurate comparison between different TA schemes, we fix all these factors and focus on the impact of TA.

A. Network and Routing Models

In our simulations, we consider the two-level hierarchical network shown in Figure 10, which consists of 22 nodes that are divided into 6 PGs. For each experiment, the QoS parameters associated with each link are generated randomly. Each link is bidirectional and asymmetric with either single or multiple QoS parameters depending on the experiment. The source and destination of a connection request are randomly chosen from Nodes 1, 2, and 3. The QoS requirements of a connection request are generated randomly. Source nodes determine the entire route to any destination using their knowledge of the network. In order to make routing decisions only depend on the aggregated information and not on other factors such as the traffic load, we assume that the true state of the network does not change after TA. In this way, we can measure how accurate a TA scheme represents the true state of the network. After a route for a connection request is determined at the source node, the simulation program checks whether this route is acceptable according to the exact state of the network. If it is acceptable, then TA scheme has resulted in a correct routing decision. Otherwise, the aggregated information is not accurate and will result in crankback.

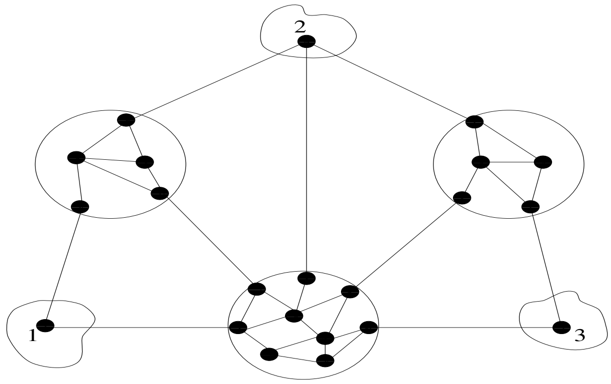


Fig. 10. Network topology used in simulations.

Another important factor is the path selection algorithm. A number of path selection algorithms are available for QoS-based routing [17]. For experiments with single QoS parameter (e.g., delay), we use Dijkstra's shortest path algorithm [9]. For experiments with multiple QoS parameters (e.g., bandwidth and delay), we use the shortest-distance path algorithm which is presented in [20] as a centralized routing algorithm. This algorithm simply prunes all links that do not meet the bandwidth requirement, and then applies Dijkstra's shortest path algorithm to find a path that meets the delay requirement of the connection request.

B. Performance Measures

In order to compare TA methods, we use the *success rate* of feasible connection requests, and the *crankback rate* per connection request. A connection request is *feasible* if it is admissible based on the exact state of the network. Using the aggregated information, the source node may or may not find a route for a feasible connection request. If it finds a route and this route is realized, we count this as a *realized connection request*. If the source node generates a route which cannot be realized, we count this as a *crankbacked connection request*. Formally,

$$success\ rate = \frac{Total\ number\ of\ realized\ requests}{Total\ number\ of\ feasible\ requests}$$

$$crankback\ rate = \frac{Total\ number\ of\ crankbacked\ requests}{Total\ number\ of\ requests}$$

In most cases, there is a conflict between these two measures. Increasing the success rate would typically results in a higher crankback rate. While a high crankback rate may increase call establishment times and waste some resources, significantly increased success rate allows the network to accept several connection requests, satisfy several users, and utilize the network resources. Although the trade-off between the success rate and the crankback rate is not quantified, we believe that a small increase in crankback rate is acceptable if the success rate increases significantly.

C. Single QoS Parameter

In this section, we compare various TA schemes assuming each link is associated with a delay value. The delay over each link is randomly chosen from a uniform distribution over [5,45(ms)]. The delay requirement of a connection request is uniformly distributed over [100,200(ms)]. Each experiment is repeated 30 times, each time with a different seed for the random number generator. In each run, 23966 connection requests are generated, 22720 of them are feasible, on average.

C.1 Source-Oriented Simple-Node Versus Conventional Simple-Node

The performances of the source-oriented and conventional simple-node schemes are contrasted in Figure 11. The source-oriented approach significantly improves the success rate over the conventional approach by about 25%. The crankback rate slightly increases, but it is overshadowed by the improvement in the success rate. Note that both versions of the simple-node have the same complexity of $\mathcal{O}(1)$.

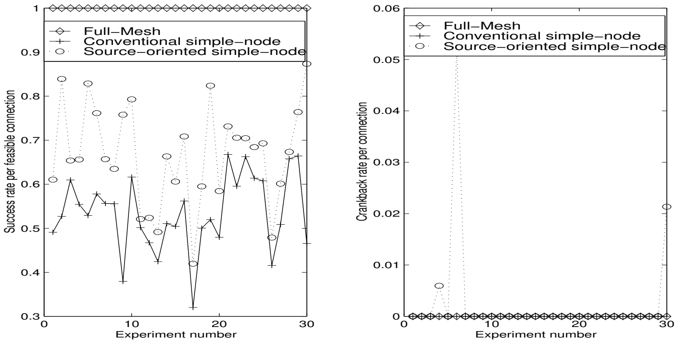


Fig. 11. Performance of two simple-node schemes and full-mesh under single QoS parameter.

C.2 Source-Oriented Star Versus Conventional Star

In the conventional star scheme, the star is mapped into a complex-node representation and advertised to all neighbors. In the source-oriented star scheme, a different quasi-star with the same star is mapped to complex-node representations and advertised to the corresponding neighbors. The performances of these two schemes and of the full-mesh are shown in Figure 12. The source-oriented approach improves the success rate by about 10% over the conventional approach. In addition, the source-oriented approach significantly reduces the crankback rate of the conventional approach about 90%. Based on these two performance measures, the source oriented star scheme is better than the conventional star scheme. Note that the source-oriented star advertises extra $M-1$ logical links (a quasi-star) in practice although the complexity of both versions is represented by $\mathcal{O}(M)$, where M is the number of border nodes.

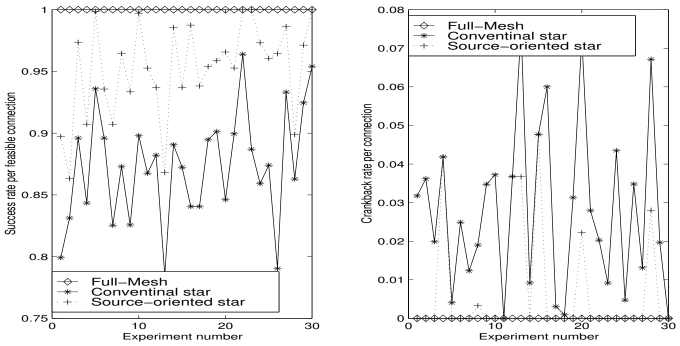


Fig. 12. Performance of two star schemes and full-mesh under single QoS parameter.

D. *Multiple QoS Parameters*

In this section, we compare TA schemes based on multiple QoS parameters, namely, bandwidth and delay. The available bandwidth and delay over each link are randomly chosen from uniform distributions over [1,10(Mbps)] and [5,45(ms)], respectively. The bandwidth and delay requirements of a connection request are uniformly distributed over [0.1,10(Mbps)] and [150,200(ms)], respectively. Again each experiment is repeated 30 times. In each experiment, 23984 connection requests are generated, 9946 of them are feasible, on average.

D.1 Finding Logical Links

In this section, we compare our two new approaches to obtaining the logical links (CSPA and Modified-MPPBCA), and the conventional ones (MPPBCA and MPPWCA). We assume that logical links are determined according to these approaches and all these logical links (i.e., the full-mesh) are advertised. The success and crankback rates of these approaches are shown in Figure 13. The modified-MPPBCA has slightly better success rate than the conventional MPPBCA. Since the modified-MPPBCA excludes part of Region II, it also reduces the crankback rate of the conventional MPPBCA by about 25%. The crankback rate can be further reduced by using the conservative estimation of $total_{s\_factor}$ based on maximum excluded QoS parameter instead of its aggressively estimated value. However, this conservative estimate may involve some degradation in the success rate. CSPA performs as good as MPPBCA in terms of success rate without causing any crankback. Thus, CSPA is preferable over MPPBCA. The success rate of MPPWCA is significantly less than the success rates of other approaches, but it has a zero crankback rate.

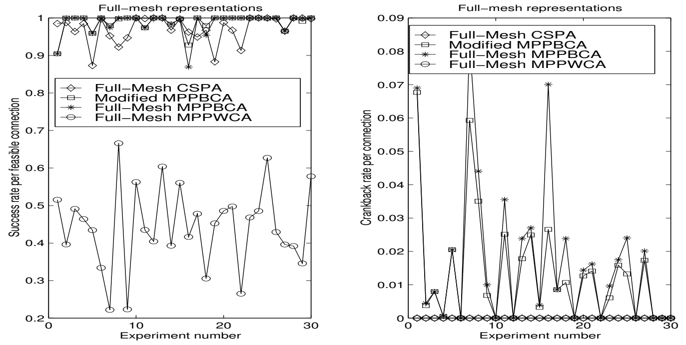


Fig. 13. Performance of various approaches for computing the weights of the logical links under multiple QoS parameters.

D.2 Source-Oriented Simple-Node Versus Conventional Simple-Node

After constructing the full-mesh using CSPA, the success and crankback rates of the conventional and source-oriented simple-nodes are shown in Figure 14. The source-oriented simple-node increases the success rate by about 75% over the conventional one. The crankback rate slightly increases. However, the increase in crankback rate might be acceptable because of the significant increase in the success rate. We also compare both versions of the simple-node scheme when the underlying full-mesh is generated by using MPPWCA and MPPBCA. In all cases, the source-oriented approach significantly improves the success rate over the conventional approach, while also increasing crankback rate by an acceptable amount. Furthermore, the source-oriented simple-node improves the routing performance over the conventional simple-node for TA under multiple QoS parameters.

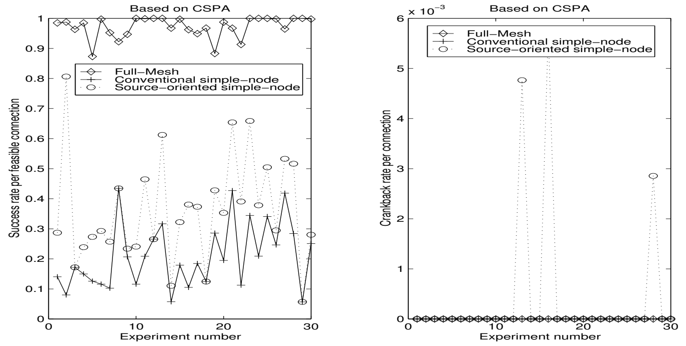


Fig. 14. Performance of two simple-node schemes and full-mesh under multiple QoS parameters.

D.3 Source-Oriented Star Versus Conventional Star

The success and crankback rates of the conventional and source-oriented stars are shown in Figure 15. The source-oriented star improves the success rate over the conventional star by about 10%. In addition, the source-oriented star decreases the crankback rate by about 60%. Based on these two performance measure, the source oriented star is better than the conventional star for TA under multiple QoS parameters.

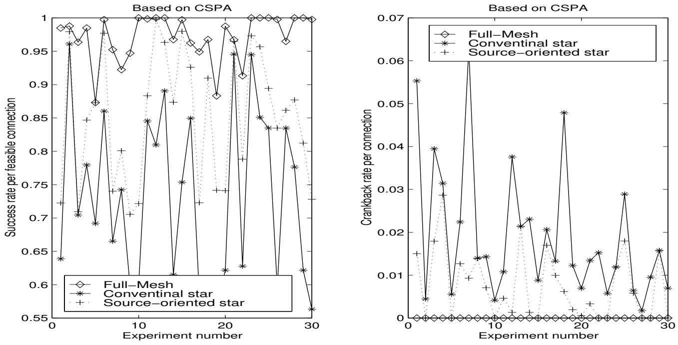


Fig. 15. Performance of two star schemes and full-mesh under multiple QoS parameters.

VI. Conclusion and Future Work

The contributions of this paper are twofold. First, we proposed the source-oriented TA approach for scalable and efficient QoS-based routing in ATM networks. The intuition behind this approach is to provide source nodes with the relevant topological information that is used in the path selection process. We integrated this approach into three new TA schemes: unified quasi-stars, source-oriented simple-node, and source-oriented star. These schemes provide different tradeoffs between compaction and accuracy. The unified quasi-stars scheme is a better alternative to the conventional advertisement of a full-mesh since it provides the same accuracy with less advertisement overhead. For further reduction in the advertisement overhead, we introduced the source-oriented simple-node and star schemes. Using simulation, we showed that these schemes provide better performance than their conventional counterparts.

Second, we introduced two new approaches (CSPA and modified-MPPBCA) for the determination of the QoS values of a logical link under multiple QoS parameters. We studied the performance of these approaches by simulation. Modified-MPPBCA slightly improves the success rate over the conventional MPPBCA. It also decreases the crankback rate. Further reduction in the crankback rate can be achieved by using the conservative estimation of $total_{s\_factor}$. CSPA performs as good as MPPBCA without causing crankbacks; thus, it is preferred over MPPBCA. In the simulation, we used these approaches with the source-oriented simple-node and star schemes for TA under multiple QoS parameters. Based on our simulation results, the source-oriented schemes also perform better than their conventional counterparts under multiple QoS parameters.

Our simulation results have been intentionally obtained for a "static" scenario in which the state of the network remains the same after TA. In practice, the state of the network dynamically changes for some parameters such as bandwidth. Because of latencies and periodic advertisements, outside nodes may not receive the true state of a PG even if the full-mesh of this PG is advertised. Thus, not only the current state of a PG must be taken into account when this PG is aggregated, but also the expected fluctuations in the state of this PG. We plan to include the expected dynamic changes in the state of a PG into TA methods to minimize the number of wrong routing decisions made by source nodes.

References

[1] C. Alaettinoglu and A.U. Shankar. The viewserver hierarchy for interdomain routing: Protocols and evaluation. *IEEE Journal on Selected Areas in Communications*, 13(8):1396–1410, Oct 1995.

[2] I. Althofer et al. On sparse spanners of weighted graphs. *Discrete & Computational Geometry*, 9(1):81–100, 1993.

[3] G. Apostolopoulos, R. Guerin, S. Kamat, and S. K. Tripathi. Quality of service based routing: A performance perspective. In *ACM SIGCOMM'98*, Vancouver, British Columbia, Canada, August-September 1998.

[4] B. Awerbuch, Y. Du, Khan. B., and Y. Shavitt. Routing through teranode networks with topology aggregation. In *Proceedings of ISCC'98*, pages 406–412. IEEE, 1998.

[5] B. Awerbuch and Y. Shavitt. Topology aggregation for directed graph. In *Proceedings of ISCC'98*, pages 47 –52. IEEE, 1998.

[6] J. Behrens and J.J. Garcia-Luna-Aceves. Hierarchical routing using link vectors. In *Proceedings of the INFOCOM'98 Conference*, volume 2, pages 702–710. IEEE, March-April 1998.

[7] K. R. Bhutani, A. Battou, and B. Khan. Two approaches for aggregation of peer group topology in hierarchical PNNI networks. The Catholic Uni. of America. Dept. of Mathematics, Washington D.C., June 1998. (Private Communications).

[8] D. Clark et al. Strategic directions in networks and telecommunications. *ACM Computing Surveys*, 28(4):579–690, 1996.

[9] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. The MIT press and McGraw-Hill book company, sixteenth edition, 1996.

[10] The ATM Forum. Private network-to-network interface specification version 1.0 (pnni 1.0), March 1996. af-pnni-0055.000.

[11] J.J. Garcia-Luna-Aceves and J. Behrens. Distributed, scalable routing based on vectors of link states. *IEEE Journal on Selected Areas in Communications*, 13(8):1383–1395, Oct 1995.

[12] R. Guerin and A. Orda. QoS-based routing in networks with inaccurate information: Theory and algorithms. In *Proceedings of the INFOCOM'97 Conference*, pages 75–83. IEEE, 1997.

[13] L. Guo and I. Matta. On state aggregation for scalable QoS routing. In *Proceedings of the ATM Workshop'98*, pages 306–314. IEEE, May 1998.

[14] W. C. Lee. Spanning tree method for link state aggregation in large communication networks. In *Proceedings of the INFOCOM'95 Conference*, pages 297–302. IEEE, 1995.

[15] W. C. Lee. Topology aggregation for hierarchical routing in ATM networks. In *ACM SIGCOMM'95, Computer Communications Review*, pages 82–92, 1995.

[16] D. H. Lorenz and A. Orda. QoS routing in networks with uncertain parameters. In *Proceedings of the INFOCOM'98 Conference*, volume 1, pages 3–10. IEEE, March-April 1998.

[17] Q. Ma and P. Steenkiste. On path selection for traffic with bandwidth guarantees. In *Proceedings of IEEE International Conference on Network Protocols*, pages 191 –202, 1997.

[18] W. T. Tsai, Ramamoorthy C.V., W. K. Tsai, and Nishiguchi O. An adaptive hierarchical routing protocol. *IEEE Transactions on Computers*, 38(8):1059–1075, August 1989.

[19] P. F. Tsuchiya. The landmark hierarchy: A new hierarchy for routing in very large networks. *ACM, Computer Communications Review*, 18(4):35–42, August 1988.

[20] Z. Wang and J. Crowcroft. Bandwidth-delay based routing algorithms. In *Proceedings of the GLOBECOM'95 Conference*, volume 3, pages 2129–2133. IEEE, Nov. 1995.