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

Turgay Korkmaz
and
Marwan Krunz

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

In this paper, we investigate the problem of topology aggregation (TA) for scalable, QoS-based routing in hierarchical networks. TA is the process of summarizing the topological information of a subset of network elements. This summary is flooded throughout the network and 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 information relevant 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 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.

Categories and Subject Descriptors: C.2.2 [**Computer-communication networks**]: Network Protocols—*QoS Routing*; C.2.6 [**Computer-communication networks**]: Internetworking

Additional Key Words and Phrases: QoS-based routing, topology aggregation, PNNI, scalable routing, ATM networks

---

Part of this paper was presented at the *Seventh International Workshop on Quality of Service (IWQoS '99)*, London, May 1999.

1. INTRODUCTION

State dependent, quality-of-service (QoS) based routing protocols necessitate the provisioning of scalable routing solutions that take into account the QoS requirements of prospective connections as well as the available network resources. Examples of such protocols are the Private Network-to-Network Interface (PNNI) [Forum 1996] of the ATM Forum and the QoS-enhanced OSPF protocol [Moy 1998; Apostolopoulos et al. 1998], both of which are link-state and hierarchical. Henceforth, we continue our discussion in the context of PNNI, although our ideas can be applied (with modifications) to the OSPF protocol [Moy 1998]. The PNNI protocol 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 network state. 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 a hierarchical routing approach in which nodal and link-state information is summarized at multiple levels in the hierarchy before being flooded throughout the network. The summarization 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 level in the hierarchy. Link aggregation refers to the process of representing a set of parallel links between two PGs 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 PGs, 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-

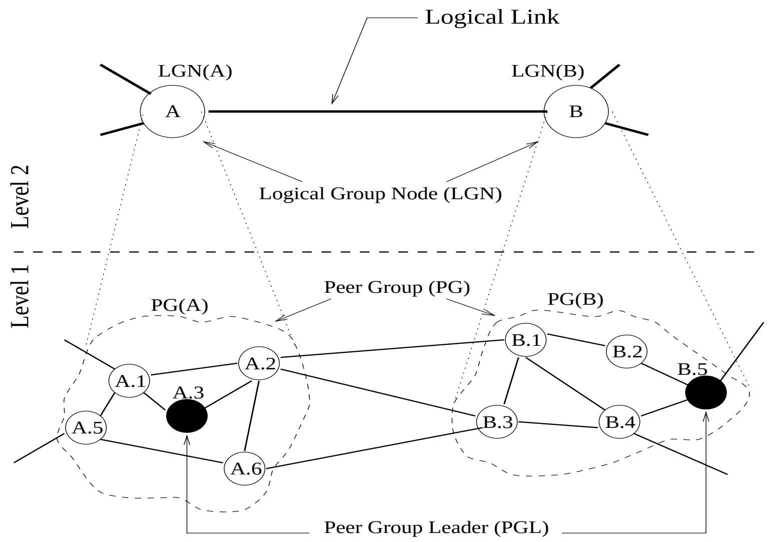


Fig. 1. Nodal and link aggregation in PNNI.

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 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.

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 [Lee 1995b]). 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. 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 pruning several links of the full mesh and mapping it into a more compact topology, such as a symmetric-node (simple-node), a star [Lee 1995b], a minimum spanning tree [Lee 1995a], a t-spanner [Althofer et al. 1993], or minimum equivalent sub-spanner [Lee 1999]. 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 problem, some researchers proposed new TA approaches that minimize the average distortion (i.e., lossyness) in a least-square sense [Bhutani et al. ions; Awerbuch and Shavitt 1998] or that maximize the precision using linear programming [Iwata et al. 1998]. In [Hao and Zegura 2000] the authors proposed two approaches. One of them (called hybrid approach) advertises relatively static information (e.g., hop count) less frequently while advertising highly dynamic information (e.g., bandwidth) more frequently and in less detail. The other approach determines the most used paths based on statistical observation and gives more weight to these paths during the next aggregation. The effects of several TA schemes on routing performance have been studied by simulation [Awerbuch et al. 1998; Awerbuch et al. 1998; Guo and Matta 1998; Hao et al. 1998]. In [Ragozini et al. 1999] the authors proposed an analytical model for evaluating the impact of aggregation under the simple-node method. In [Van Mieghem 1997; Rougier et al. 2000] the authors investigated how to cluster a given topology into PGs under some optimization criteria. In [Guerin and Orda 1999; Lorenz and Orda 1998] the authors presented several heuristic algorithms for route selection in the presence of inaccurate topological information, including inaccuracies that are caused by TA. In [Basturk and Stirpe 1998] the authors investigated efficient dissemination of topology information in PNNI. There have been several other proposals for hierarchical routing in the literature [Tsuchiya 1988; Tsai et al. 1989; Alaettinoglu and Shankar 1995; Behrens and Garcia-Luna-Aceves 1998; Garcia-Luna-Aceves and Behrens 1995]. These studies mainly focus on address aggregation and reachability rather than the aggregation of state information.

Contributions and Organization of the Paper

The contributions of this paper are twofold. First, we introduce a novel, source-oriented approach to TA, in which only relevant topological information is advertised. Relevance is defined relative to the *source* nodes that compute the tentative routes for connection requests. Since the relevant information 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 with 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 these schemes achieve better performance than their conventional counterparts. Our second contribution is in 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, in the construction of a full-mesh of border nodes. Extensive simulations are used to evaluate the performance of our schemes and contrast them with previously proposed schemes.

The rest of the paper is organized as follows. In Section 2, we start by describing the conventional TA approach and then introduce a generic source-oriented TA methodology. Based on this methodology, three different source-oriented TA schemes are presented in Section 3. Two strategies for obtaining the weights of a logical link under multiple QoS parameters are given in Section 4. In Section 5 we evaluate the performance of our schemes and contrast them with conventional TA schemes. Finally, the paper is concluded in Section 6.

2. TOPOLOGY AGGREGATION

In this section we describe the conventional and the proposed source-oriented TA approaches. For illustration purposes, we use the two-level hierarchical topology shown in Figure 2, which consists of ten nodes clustered into five PGs. We focus on the aggregation of the central PG. Links are assumed bidirectional and asymmetric.

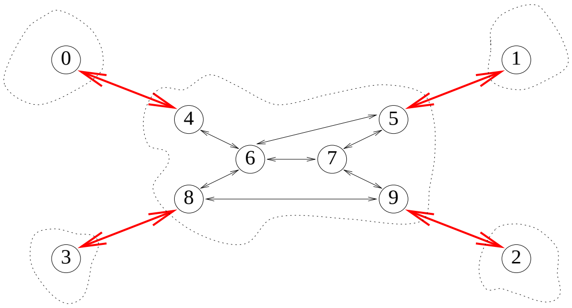


Fig. 2. Two-level hierarchical network topology.

2.1 Conventional Approach

According to the conventional TA approach, the central PG is first mapped into a full-mesh of 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 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

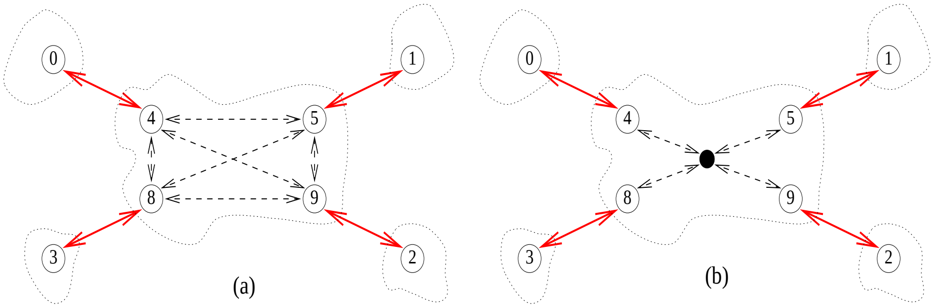


Fig. 3. Conventional approach to aggregating the central PG (a) full-mesh generation; (b) graph reduction.

the routing performance. Note that the same information about the central PG is being advertised throughout various border nodes irrespective of the relevance of this information to the route selection mechanism at these nodes.

2.2 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. If Node 0 needs to compute a route for a connection request that traverses the central PG, it only needs to know the costs of the logical links from Node 4 to every other border node. The other logical links 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 (see Figure 4).

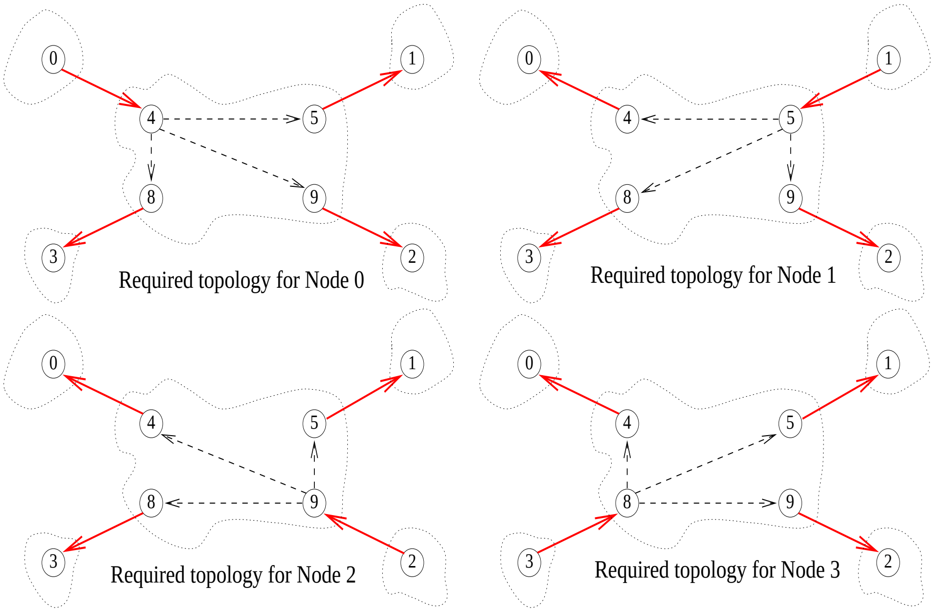


Fig. 4. Required topology information for Nodes 0, 1, 2, and 3.

Advertising different partitions of the full-mesh to different neighbors based on their needs results in better compaction than the full-mesh advertisement at no loss in 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.

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 there exists a direct link between Nodes 0 and 3, then Node 0 needs to receive two different partitions of the central PG, which are advertised through nodes 4 and 8, respectively. Therefore, the overhead of 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 a 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.

3. 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 are aimed at dense topologies. First, let's define a new topology called *quasi-star*, which is a star-like topology whose center 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, the PGL constructs a full-mesh of border nodes and partitions it into M quasi-stars. It then associates each quasi-star with one border node and maps it into a complex-node representation with different $M-1$ bypasses. The M distinct representations are advertised to the outside through the corresponding border nodes. An outside node uniquely recognizes a given representation based on the identity of its nucleus and any one of the bypasses. If the node receives different quasi-star representations that were generated at the same time (e.g., carry the same sequence number), it combines them in its database and forwards the newly received one to all neighbors except the one from which the advertisement was received. If the node receives a subsequent advertisement of an existing quasi-star, it checks the sequence number (SN) in this advertisement. If this SN is smaller than or equal to the SN of the currently stored advertisement, the newly received advertisement is discarded. Otherwise, the new advertisement is stored in place of the older one and is forwarded to all neighbors as described before. Implementing Scheme 1 requires minor changes to the standard operation of PNNI, namely that a node needs to combine different quasi-stars in its database.

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 needs only to advertise this quasi-star (i.e., compact triggered updates). In contrast, the complete full-mesh would have to be advertised in the conventional approach.

The advertisement overhead in Scheme 1 may still be an issue, as it depends on the density of the inter-PG connectivity. For dense topologies, further reduction in this overhead is desirable at the expense of some lossyness. One way to do that is to combine and forward the first received τ different quasi-stars and discard subsequent ones, where $1 \leq \tau \leq M - 1$. We now provide two such schemes for which $\tau = 1$.

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 [Ma and Steenkiste 1997; Guo and Matta 1998]. 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 eventually receive different advertisements from the same PG. The first received advertisement is likely to have traversed one of the shortest paths to this node. The reverse path is also expected to be one of the shortest ones since propagation delay is symmetric and it is the dominant delay component in high-speed networks [Clark et al. 1996]. More accurate information can be acquired on the short paths by storing and forwarding only 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 per QoS parameter. This value, typically the diameter of the PG [Lee 1995b], is advertised to all neighbors of the PG. We now present a source-oriented version of this scheme. Suppose the PG contains M border nodes. Let d_{ij} be the distance from node i to node j, for all i and j. To aggregate a PG, the PGL constructs a full-mesh of border nodes and selects the cost of the *worst* path from a given border node to every other border node. For example, for the delay parameter the PGL computes

$$d_{max}(i) \overset{\triangle}{=} \max_{j} d_{ij} \text{ for } i = 1, \ldots, M$$

The M different values $d_{max}(i)$, $i = 1, 2 \ldots, M$, are individually advertised to the outside through the corresponding border nodes. These 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 distinct advertisement is $\mathcal{O}(1)$, as in the conventional simple-node scheme.

Scheme 3: Source-oriented star

One common TA approach is based on the asymmetric star topology [Lee 1995b]. 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 [Guo and Matta 1998; Awerbuch et al. 1998; Bhutani et al. ions]. Without loss of generality, we consider the average-case approach. After constructing a full-mesh of border nodes, the PGL compacts this full-mesh into a star by computing the weights of links from every border node to the fictitious nucleus, and vice versa. More formally, the PGL computes the following (average) distances:

$$\begin{aligned} d_{if} &= \frac{1}{M-1} \sum_{j=1, j\neq i}^{M} d_{ij} \quad \text{for } i = 1, 2, \ldots, M \\ d_{fj} &= \frac{1}{M-1} \sum_{i=1, i\neq j}^{M} d_{ij} \quad \text{for } j = 1, 2, \ldots, M \end{aligned}$$

The advertisement 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 [Forum 1996; Lee 1995b]. 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 determines the star representation of the full-mesh, as described above. It then combines each quasi-star with the common star and constructs M distinct complex-node representations of the same PG. The PGL advertises these complex-node representations with the same SN to the outside through corresponding border nodes. When an outside node receives a complex-node representation for the first time, it stores it and forwards it to its neighbors except the one from which the advertisement received. Subsequent advertisements with equal or smaller SNs are discarded. An example of this scheme is shown in Figure 5, which depicts the state information available at nodes 0 and 1 regarding the central PG.

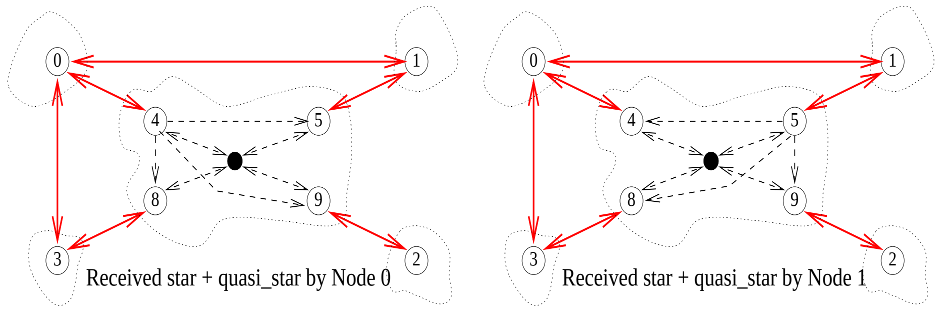


Fig. 5. TA of the central PG from the view point of Nodes 0 and 1.

4. AGGREGATION UNDER MULTIPLE QOS PARAMETERS

One fundamental step in TA is the assignment of weights 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 graph. 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. 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 enhanced strategies.

4.1 Logical Links with Multiple Parameters

Figure 6(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. These four paths are to be represented 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 7. In this figure, the costs of the four paths are shown in circles. 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 exact admission region, i.e., a connection request whose QoS requirements fall in this region will be accepted. The main issue here is how to define and determine the parameters of a logical link so that the exact admissible region is represented as accurately as possible. There is a tradeoff between the accuracy of the representation and the number of parameters associated with a logical link. For example, conventional approaches (reviewed in the next section) associate a logical link with a single set of QoS values (i.e., a single point in the parameter space). However, this single point is not enough to represent the exact shape of the admissible region. As a matter of

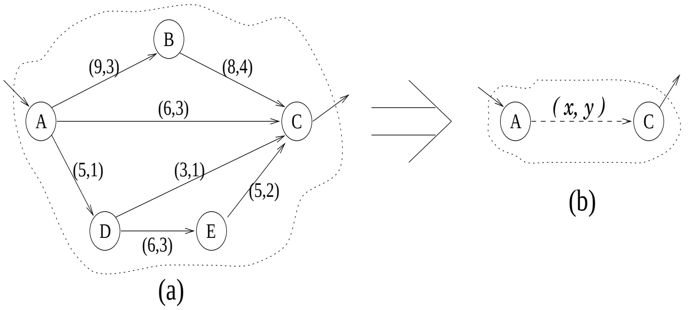


Fig. 6. A PG in which each link has two parameters (BW, delay).

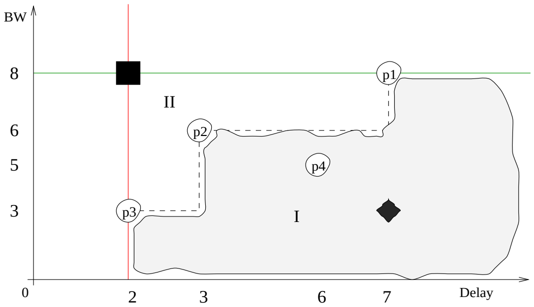


Fig. 7. Partitioning the QoS-parameter space based on physical paths.

fact, the exact shape requires specifying the costs of all *non-dominated* paths that bound the admissible region (e.g., p_1, p_2, and p_3 in Figure 7), which is not a scalable solution.

4.2 Conventional Approaches

Existing approaches find a single point in the parameter space and use the QoS values of this point as the parameters of the logical link. There are several strategies for determining such a single point.

- **Single-Path-Parameters Approach** (SPPA): 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 [Lee 1995b]. The key issue here 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 exact 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 7), then the admissible region as seen by an outside node is restricted to the area that lies to the bottom right of p3. SPPA may cause 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 of all paths to the logical link (i.e., the black box in Figure 7). It is an aggressive approach since it is quite possible that none of the physical paths can simultaneously support all the advertised QoS values [Lee 1995b]. 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 a high probability of crankback.
- **Multiple-Path-Parameters-Worst-Case Approach** (MPPWCA): This approach finds the worst value for each QoS parameter from the best paths with respect to other QoS parameters (i.e., the diamond in Figure 7), and assigns these values to the logical link. In this approach, the utilization is expected to be relatively low because many requests will be unnecessarily blocked or rerouted. However, we later show that this approach is less sensitive to network dynamics than the other two approaches.

4.3 Two New Strategies

The above three approaches can be overly aggressive or conservative in approximating the actual admissible region, resulting in excessive crankbacks or in significant underutilization of network resources, respectively. In this section, we provide two new approaches that give better representation of the admissible region. Before describing these two strategies, we define a new parameter called *stretch factor* (s_factor). The minimization of s_factor is in the core of both strategies.

Stretch Factor (s_factor). Let $P = \{p_1(Q_1^1, ..., Q_1^K), ..., p_r(Q_r^1, ..., Q_r^K)\}$ be the set of r 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) 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

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

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

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

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* as

$$s\_factor(p_i) \triangleq \left(\sum_{k=1}^{L} \frac{Best_{Q^k}}{Q_i^k} + \sum_{k=L+1}^{K} \frac{Q_i^k}{Best_{Q^k}} \right) \tag{1}$$

The strech factor measures the "distance" between the cost of a physical path and the cost of an *ideal* (often, nonexistent) path that is optimal with respect to every QoS parameter (i.e., the black square in Figure 7). We now provide a polynomial-time algorithm that finds a path with the minimum s_factor in the case of one attribute and any number of additive metrics.

Consider a PG that is represented by a directed graph $G = (V, E)$, where V is the set of nodes and E is the set of links. Suppose each link $(i, j) \in E$ is associated with K QoS parameters $Q^1(i,j), Q^2(i,j), \ldots, Q^K(i,j)$, which consist of one attribute (e.g., bandwidth) and $K - 1$ metrics. The problem is to find a path p between a pair of border nodes (e.g., s and t) such that p has the minimum s_factor among all paths from s to t. In this case, the s_factor of path p is given by

$$s\_factor(p) \triangleq \left(\frac{Best_{Q^1}}{Q^1(p)} + \sum_{k=2}^{K} \frac{Q^k(p)}{Best_{Q^k}} \right) \tag{2}$$

where $Q^k(p)$ is the k^{th} QoS parameter of p. To find a path with the minimum s_factor, we propose the algorithm in Figure 8. The algorithm first computes $Best_{Q^k}$ for all k's. For each additive metric $k = 2, 3, \ldots, K$, $Best_{Q^k}$ is computed by using Dijkstra's shortest path algorithm [Cormen et al. 1996]. For the attribute ($k = 1$), finding $Best_{Q^1}$ is known as the maximum capacity path problem and can be computed by a simple modification of Dijkstra's algorithm [Ahuja et al. 1993]. In the second step, the algorithm combines the additive parameters as a

```
Minimum_$s\_factor(G(V,E),s,t)$
begin
1  Compute $Best_{Q^k}$ from $s$ to $t$ for $k = 1,2,\ldots,K$
2  Associate each link $e \in E$ with one weight: $w(e)$
3     $w(e) = \sum_{k=2}^{K} \frac{Q^k(e)}{Best_{Q^k}}$      /* $w(e)$ is a metric */
4  Find the shortest path $q$ w.r.t. $w(e)$
5  $s\_factor\_min = s\_factor(q)$
6  $p = q$
7  while $(Q^1(q) < Best_{Q^1})$ do
8     Prune every link $e \in E$ for which $Q^1(e) \le Q^1(q)$
9     Find the shortest path $q$ w.r.t. $w(e)$ in the reduced graph
10    if $s\_factor(q) < s\_factor\_min$ then
11       $s\_factor\_min = s\_factor(q)$
12       $p = q$
13    end if
14 end while
15 return $p$ and $s\_factor\_min$
end
```

Fig. 8. An algorithm for finding a path p with minimum s_factor.

single metric by computing the sum of their individual stretches. The algorithm then considers all *non-dominated* paths that bound the exact admission region. For example, in the case of bandwidth and delay, these paths correspond to the (BW, delay) points on the staircase that bounds the feasibility region (p_1, p_2, and p_3 in Figure 7). Since the algorithm considers all the paths that bound the feasibility region, its optimality is intuitively correct and the proof is straightforward. The computational complexity of this algorithm is discussed next.

In the first 6 lines, the algorithm computes $Best_{Q^k}$ for $k = 1, 2, \ldots, K$, associates a new weight $w(e)$ with each link, and finds the shortest path w.r.t. $w(e)$. To find $Best_{Q^k}$ and the shortest path w.r.t. $w(e)$, the algorithm uses Dijkstra's algorithm whose complexity is $\mathcal{O}(n \log n + m)$. The algorithm then enters the main loop (lines 7-14) in which some links are pruned and a shortest path is found. The algorithm iterates this loop b times, where b is the number of distinct path costs for the attribute considered by the algorithm. As a result, the complexity of the algorithm is $\mathcal{O}((K + b) * (n \log n + m))$. In the worst-case where each path has a different Q^1 value, b is equal to the number of links (i.e., m). Since $b \le m$, the worst-case complexity of the algorithm is clearly polynomial. However, its average complexity is observed to be much less than that since, on average, $b \ll m$. We have conducted simulations on random graphs to investigate the average and maximum observed values of b. The results are summarized in Table 1. The reason for the big difference between the average m and average b has to do with the fact that for an attribute (e.g., bandwidth), the cost of an end-to-end path is given by the minimum link value along that path. So, for example, all paths that share a bottleneck link will have the same bandwidth cost, resulting in very few path costs in the network. Note that the above m, n, and b parameters are for a single PG. These parameters are much smaller in value than their counterparts for the whole network [Van Mieghem

Table 1. Average and maximum values of b in random graphs with 50, 100, and 200 nodes.

# of Nodes	Average m	Average b	Maximum b
50	161	1.8	12
100	597	4.6	19
200	920	4.5	21

1997], making the $\mathcal{O}((K+b)*(n\log n+m))$ complexity quite affordable in practical networks (where PGs have up to 100 or 200 nodes). The value of b can be further reduced if necessary, by bandwidth quantization, resulting in approximate value for the s_factor that is obtained with reasonable computational complexity.

Strategy 1: Closest-Single-Path Approach (CSPA). The idea behind CSPA is to find a single physical path whose cost is the "closest" to the MPPBCA point and to associate the QoS values of this path with the logical link. Here we define closeness in terms of the s_factor. Therefore, CSPA uses the algorithm in Figure 8 to find a single path with the minimum s_factor (i.e., the closest path to the best QoS values). For the example in Figure 6, p_2 has the minimum stretch factor, so its QoS values $(6,3)$ are assigned to the logical link. Minimizing s_factor is a reasonable approach under the assumption that all QoS parameters have the same importance. However, the QoS parameters might differ in their relative significance, depending on various factors (administrative policies, nature of applications, etc.) In such cases, our definition of the s_factor can be easily extended by assigning a weight to each term in (1) with value that reflects the relative importance of the corresponding QoS parameter. More specifically, (1) can be replaced by:

$$s\_factor(p_i) \triangleq \left(\sum_{k=1}^{L} \alpha_k \frac{Best_{Q^k}}{Q_i^k} + \sum_{k=L+1}^{K} \alpha_k \frac{Q_i^k}{Best_{Q^k}} \right)$$

where $\sum_{i=1}^{K} \alpha_i = 1$ and $\alpha_i \geq 0$ for all i. Note that the above generalization accommodates the conventional SPPA by setting one of the α's to one and the remaining α's to zero.

Strategy 2: Modified-MPPBCA. The problem with the conventional MPPBCA is that it is too aggressive, i.e., outside nodes perceive both Regions I and II in Figure 7 as admissible, and accordingly, they route connection requests through the underlying PG. However, since ultimately the requirements of a connection request that falls in Region II cannot be satisfied, this request is cranked back and a new path is searched for. To reduce the probability of crankback while maintaining high utilization of network resources, we propose to modify MPPBCA such that outside nodes can exclude part of Region II and block more infeasible routes in the first place. For this purpose, we associate with a logical link the minimum s_factor along with the best QoS values $(Best_{Q^1}, Best_{Q^2}, \ldots, Best_{Q^K})$.

We now explain how outside nodes decide to route a connection request with QoS requirements $(Req_{Q^1}, Req_{Q^2}, \ldots, Req_{Q^K})$ through a logical link. As before, we assume that the first L parameters are attributes while the remaining $K-L$ parameters are metrics. In the conventional MPPBCA, if a logical link satisfies both of the following conditions:

$$Req_{Q^k} \leq Best_{Q^k}, \quad \forall k = 1, 2, \ldots, L \tag{3}$$

$$Req_{Q^k} \geq Best_{Q^k}, \quad \forall k = L+1, L+2, \ldots, K \tag{4}$$

then an outside node assumes that there is a feasible path and thus route the request through this logical link. However, if the request falls into Region II, it will be rejected and cranked back. To remedy this situation, Modified-MPPBCA checks the following additional condition:

$$s\_factor \leq \left(\sum_{k=1}^{L} \frac{Best_{Q^k}}{Req_{Q^k}} + \sum_{k=L+1}^{K} \frac{Req_{Q^k}}{Best_{Q^k}} \right) \tag{5}$$

If this condition is satisfied along with the above two conditions, then the request is routed through the logical link. Since this condition excludes part of Region II, the routed request has better chance of being accepted than in conventional MPPBCA.

For the previous example, Figure 9 depicts how outside nodes perceive the shape of the admissible region under Modified-MPPBCA. Suppose that we have three connection requests: C_1, C_2, and C_3, (see Figure 9). Only C_3 is truly admissible. Based on the conventional MPPBCA, a source node thinks that all of the three requests are admissible. In contrast, under Modified-MPPBCA, the same source node thinks that only C_2 and C_3 are admissible since it excludes the region that C_1 belongs to. Consequently, Modified-MPPBCA causes only one crankback while the conventional MPPBCA causes two crankbacks for C_1 and C_2.

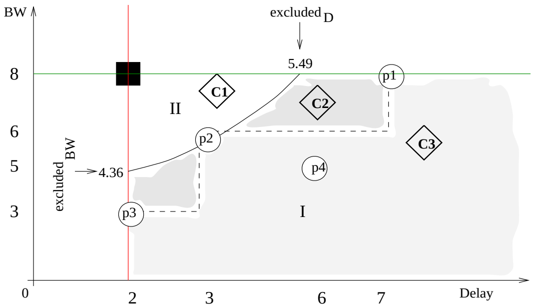


Fig. 9. Admissible region based on Modified-MPPBCA.

So far we considered a single logical link. However, a connection request can be routed over a path that contains several logical and physical links. We now show how a routing algorithm can decide whether such a path is likely to meet the end-to-end QoS requirements of a connection request. Assume that a given path p consists of l logical and physical links and that 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 equal to K. 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

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

If the following conditions

$$Req_{Q^k} \le total_{Q^k}, \quad \forall k = 1, 2, \ldots, L \tag{6}$$

$$Req_{Q^k} \ge total_{Q^k}, \quad \forall k = L+1, L+2, \ldots, K \tag{7}$$

are *simultaneously* satisfied, then based on the conventional MPPBCA, a source node decides that the path p is likely to meet the end-to-end QoS requirements of the connection request. However, since $total_{Q^k}$, $k = 1, 2, \ldots, K$, are found based on the best QoS parameters, there may not be a single physical path that meets all the requirements (i.e., the connection request may fall in Region II). To overcome this situation, Modified-MPPBCA requires the satisfaction of the following condition along with the above two conditions

$$total_{s\_factor} \le \left(\sum_{k=1}^{L} \frac{total_{Q^k}}{Req_{Q^k}} + \sum_{k=L+1}^{K} \frac{Req_{Q^k}}{total_{Q^k}} \right) \tag{8}$$

where $total_{s\_factor}$ is the s_factor of path p, which is not readily available. We now show how to compute this quantity.

For each of the K QoS parameters, we calculate the best possible path cost *after* excluding part of Region II. These new values are obtained as follows:

$$excluded_{Q^k} = \min\{\frac{Q_i^k}{(s\_factor_i - (K-1))} \mid 1 \le i \le l\} \tag{9}$$

for $k = 1, ..., L$ (the L attributes) and

$$excluded_{Q^k} = \sum_{i=1}^{l} Q_i^k (s\_factor_i - (K-1)) \tag{10}$$

for $k = L+1, ..., K$ (the $K-L$ metrics). Figure 9 shows these values for bandwidth ($excluded_{BW}$) and delay ($excluded_D$). Then, $total_{s\_factor}$ for the entire path can be *aggressively* estimated based on the minimum excluded QoS parameter as follows:

$$total_{s\_factor} = \min \left\{ \frac{total_{Q^k}}{excluded_{Q^k}}, \frac{excluded_{Q^{k'}}}{total_{Q^{k'}}} \right\} + (K-1) \tag{11}$$

where $1 \le k \le L$ and $L+1 \le k' \le K$. The purpose of the minimization in (11) is to achieve high utilization of network resources. Nonetheless, other, more conservative estimators can also be used (e.g., multiplying the minimum s_factor by a constant greater than one before assigning it to a logical link, or taking the maximum instead of the minimum in the above equation). But that comes at the expense of unnecessarily denying a larger number of connection requests, hence reducing network utilization. In the next section, we investigate the impact of aggressive versus conservative estimation of $total_{s\_factor}$.

5. PERFORMANCE EVALUATION

In this section, we compare various TA schemes with respect to the size of the aggregated information and the overhead of distributing this information. We then consider the accuracy of the aggregated information and its impact on the routing performance. 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 [Ma and Steenkiste 1997; Apostolopoulos et al. 1998]. To measure how accurate a TA scheme represents the true state of the network, we first compare TA schemes in a "static" environment where the true state of the network does not change after TA. We then conduct our comparisons using a more realistic simulation setup in which network dynamics (e.g., advertisement frequency, state updates following successful call admissions) are taken into account.

5.1 Network and Routing Models

In our simulations, we consider two two-level hierarchical network topologies. The first one (Figure 10) is a dense topology that consists of 22 nodes clustered into 6 PGs. The second one (Figure 11) is sparse and consists of 33 nodes clustered into 12 PGs. In each experiment, the link parameters are randomly generated. Each link is bidirectional and asymmetric with either one or multiple QoS parameters, depending on the experiment. The QoS requirements of a connection request are generated randomly. A source node determines the entire route to any destination based on its knowledge of the network state. In our experiments with one QoS parameter, we use Dijkstra's shortest path algorithm [Cormen et al. 1996]. Under two QoS parameters (e.g., bandwidth and delay), we use the shortest-distance path algorithm which is presented in [Wang and Crowcroft 1995; Wang and Crowcroft 1996] as a centralized QoS routing algorithm. This algorithm simply prunes all links that do not meet the bandwidth requirement and then applies Dijkstra's algorithm to find a path that meets the delay requirement of the request. In our experiments with three QoS parameters (e.g., bandwidth, delay, delay-jitter), we first prune all links that do not meet the bandwidth requirement and then we use

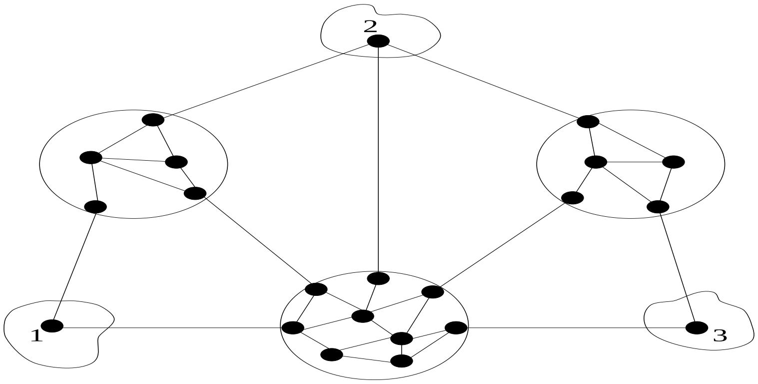


Fig. 10. First network topology used in simulations.

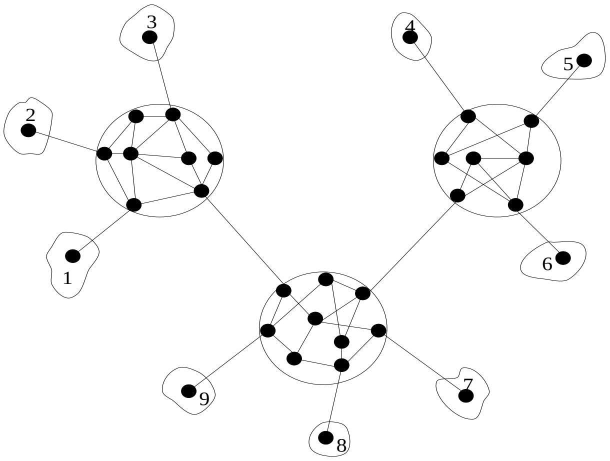


Fig. 11. Second network topology used in simulations.

Jaffe's linear approximation algorithm [Jaffe 1984] for finding a path under two additive parameters. Note that finding a path that satisfies two additive constraints is an NP-complete problem, which can only be dealt with using heuristics and approximation algorithms. Such algorithms do not always succeed in finding feasible paths, even if one exists.

5.2 Performance Measures

We contrast different TA schemes in terms of their impact on the routing performance using two measures: *success rate* (SR) and *crankback rate* (CR):

$$\mathrm{SR} = \frac{Total\ number\ of\ realized\ requests}{Total\ number\ of\ feasible\ requests}$$

$$\mathrm{CR} = \frac{Total\ number\ of\ crankbacked\ requests}{Total\ number\ of\ requests\ for\ which\ the\ algorithm\ returns\ a\ route}$$

A connection request is feasible if a path selection algorithm finds a route that meets the requirements of that request based on the exact network state. Using the aggregated information, the path selection algorithm at the source node may or may not return a route for a connection request. If it returns a route that is also feasible with respect to the exact state, then we say that the TA scheme has resulted in a correct routing decision, and that the request has been *realized.* If the returned route is infeasible, the connection request will eventually be crankbacked (this is true whether or not there exists a feasible path based on the exact state). Note that while the numerators of SR and CR are complimentary (their sum is the total number of requests for which the path selection algorithm returns a path), their denominators are not. As a result, the SR and CR are not exactly inversely proportional. For example, if the algorithm returns an infeasible path in response to a feasible connection request, both SR and CR will be negatively affected, since the algorithm failed to identify a feasible route (which exists) and it also gave a bad route. However, one can generally expect that improving the SR will have some negative impact on the CR.

5.3 Topology Advertisement Overhead

In PNNI, topology information is disseminated using PNNI Topology State Packets (PTSPs). Each PTSP has a 44-byte header. Both source-oriented and conventional simple-node schemes advertise the same amount of information in one PTSP. In the conventional star scheme, $2M$ links are advertised in one PTSP, as opposed to $3M - 1$ links in the source-oriented version (Scheme 3). As for Scheme 1, its advertisement overhead depends on the network topology. In dense topologies, this scheme advertises the same amount of aggregated information as the full-mesh. But its byte-wise overhead is higher since more PTSPs are advertised than in the full-mesh, increasing the total PTSP header overhead. In sparse topologies, since most quasi-stars need not be received by all nodes, Scheme 1 advertises less information than the full-mesh scheme, overshadowing its higher PTSP header overhead.

Using simulation, we now show how many PTSPs are advertised and the total number of bytes exchanged during one period of database synchronization. We assume that the topology database at each node is initially empty and that a PGL knows the exact topology of its own PG. Each PGL constructs the full-mesh representation of its PG and advertises it as a full-mesh or quasi-stars, depending on the aggregation scheme. In addition to the 44-byte header and the advertised QoS parameters. a PTSP may contain other fields [Forum 1996], which we ignore for simplicity. Accordingly, the size (in bytes) of a PTSP is given by $44 + (\text{number of logical links}) * K * (\text{size of a QoS parameter})$, where K is the number of QoS parameters associated with each link. For each PTSP, a 36-byte acknowledgment is used. Table 2 shows the advertisement overhead of the full-mesh and the unified quasi-stars under sparse and dense topologies when $K = 2$ and each QoS parameter is coded using 2 bytes. The table shows that the unified quasi-stars scheme has about 33% less overhead than the full-mesh scheme when the topology is sparse, but slightly higher overhead when the topology is dense. Of course, this scheme is intended for sparse topologies.

Table 2. Advertisement overhead of the full-mesh and source-oriented quasi-stars.

TA scheme	Sparse Topology (Figure 11)		Dense Topology (Figure 10)	
	Bytes Exchanged	No. of PTSPs	Bytes Exchanged	No. of PTSPs
Full-mesh	6496	47	3432	27
Unified quasi-stars	4384	47	3440	38

5.4 Routing Performance

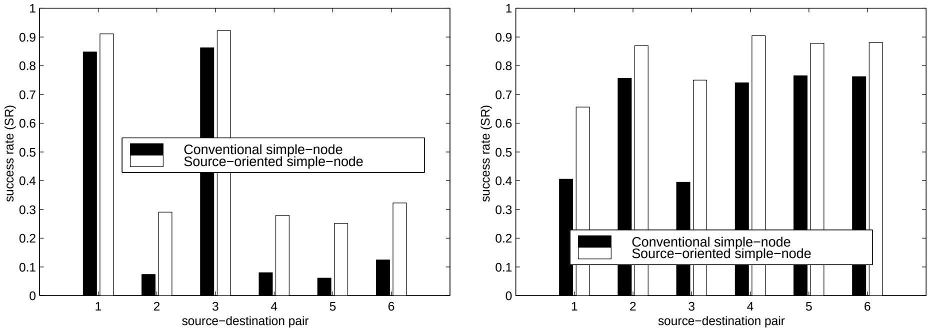
Case I: Single QoS Parameter — Static Environment. In this section, we compare various TA schemes when each link is associated with a delay value. The delay over a link is randomly chosen from a uniform distribution over the interval [5,45(ms)]. The delay requirement of a connection request is uniformly distributed over the interval [100,200(ms)] in the first topology and [150,350(ms)] in the second one. The source-destination pairs of connection requests are given in Table 3.

Table 3. Source-destination pairs used in simulations.

Source-Destination	First Topology (Figure 10)		Second Topology (Figure 11)	
Pair No.	Source Node	Destination Node	Source Node	Destination Node
1	1	2	random(1,2,3)	random(4,5,6)
2	1	3	random(1,2,3)	random(7,8,9)
3	2	1	random(4,5,6)	random(1,2,3)
4	2	3	random(4,5,6)	random(7,8,9)
5	3	1	random(7,8,9)	random(1,2,3)
6	3	2	random(7,8,9)	random(4,5,6)

Note that since the full-mesh is lossless for a single QoS parameter, it has SR = 1 and CR = 0.

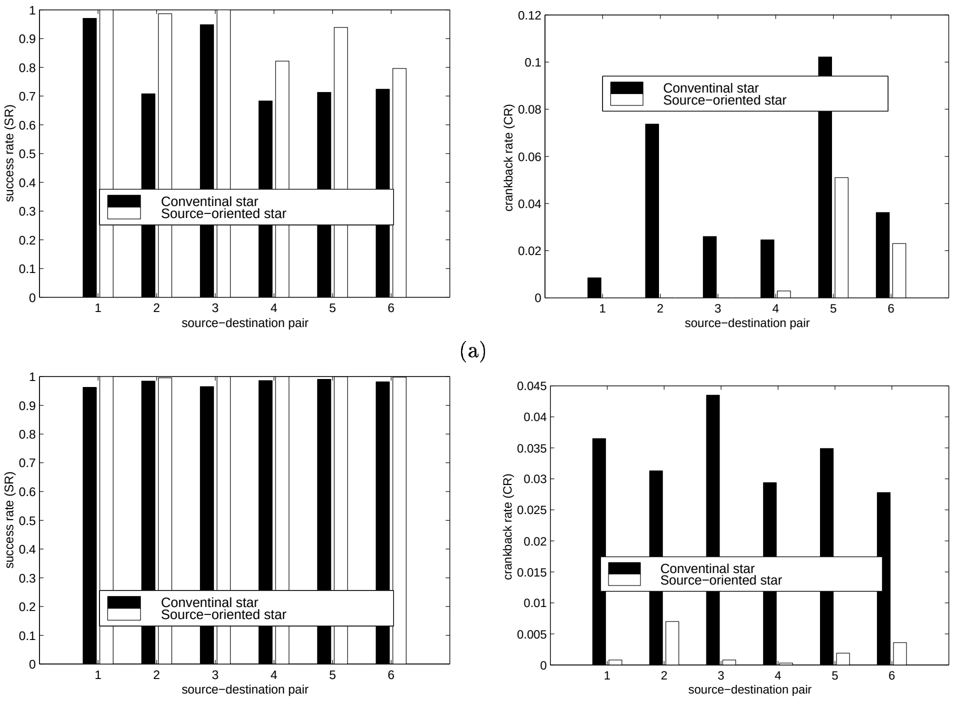
Source-Oriented Simple-Node Versus Conventional Simple-Node. Figure 12 shows the SR for the two simple-node schemes. The source-oriented approach significantly improves the SR in both dense and sparse topologies. The CR (not shown) slightly increases in the first topology, but it is overshadowed by the improvement in SR. In the second topology, because of its tree like structure, the CR is equal to zero for both schemes. Note that both versions of the simple-node have the same $\mathcal{O}(1)$ advertisement overhead.



(a) (b)

Fig. 12. Performance of two simple-node schemes under a single QoS parameter: (a) first topology, (b) second topology.

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, different quasi-stars with a common star component are mapped to complex-node representations and advertised to the corresponding neighbors. The performance of these two schemes is shown in Figure 13. The source-oriented approach improves the SR, making it close to that of the full-mesh, particularly for the sparse topology. In addition, it significantly reduces the CR over the conventional approach. Note that the source-oriented star advertises extra $M - 1$ logical links (a quasi-star) although the complexity of both schemes is $\mathcal{O}(M)$.



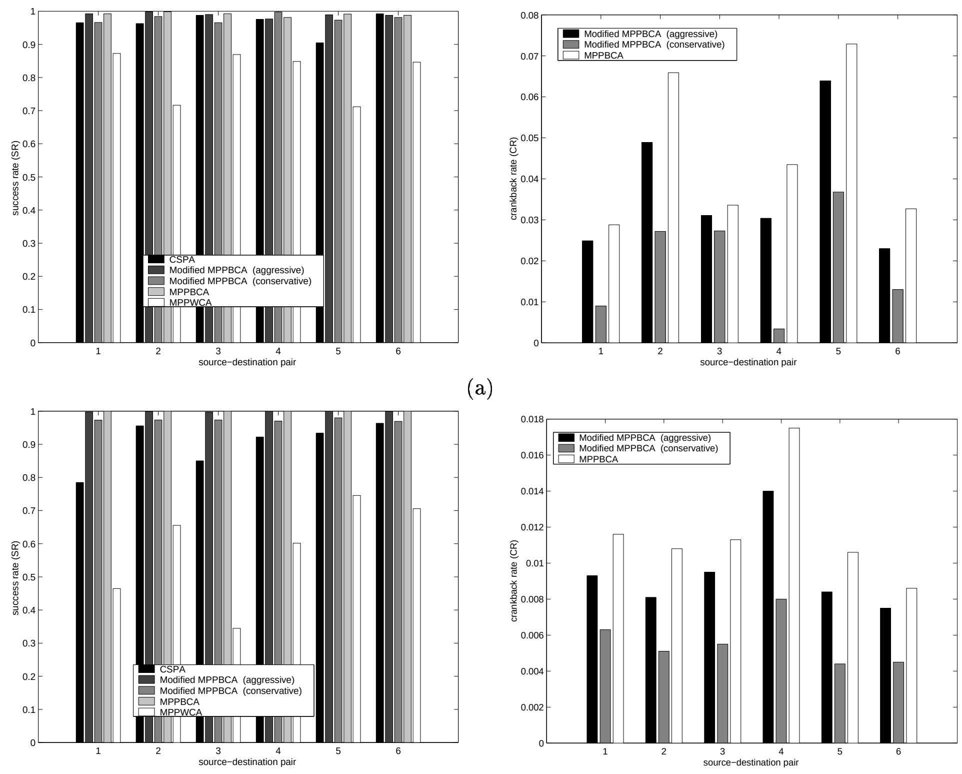
(a)

(b)

Fig. 13. Performance of two star schemes under a single QoS parameter: (a) first topology, (b) second topology.

Case II: Two QoS Parameters — Static Environment. We now consider the case when each link is associated with available bandwidth and delay values, which are randomly chosen from uniform distributions over the intervals [1,10(Mbps)] and [5,45(ms)], respectively. The bandwidth requirement of a connection request is uniformly distributed over [0.1,10(Mbps)]. The delay requirement is uniformly distributed over [150,200(ms)] in the first topology and [150,350(ms)] in the second topology. We use the same source-destination pairs of Table 3.

Finding Logical Links. We first compare the proposed CSPA and Modified-MPPBCA schemes along with the conventional ones (MPPBCA and MPPWCA). We assume that logical links are determined according to these approaches and all of the logical links of the full-mesh are advertised. The performance of these approaches is shown in Figure 14. The modified-MPPBCA with aggressive estimation of s_factor gives almost the same SR as the conventional MPPBCA. But since modified-MPPBCA excludes part of Region II, it has a lower CR. The CR is further reduced when the modified-MPPBCA is used with conservative estimation of s_factor. However, this also slightly reduces the SR when compared to the conventional MPPBCA. In terms of the SR, CSPA performs almost the same as MPPBCA when the topology is dense and slightly worse than that when the topology is sparse. However, in contrast to MPPBCA, CSPA does not cause any crankback, and hence it is preferable over MPPBCA. As for MPPWCA, its SR is significantly less than the SRs of



(a)

(b)

Fig. 14. Performance of various approaches for computing the weights of the logical links under two QoS parameters: (a) first topology, (b) second topology.

other approaches, but it has a zero CR and, more importantly, it is less sensitive to network dynamics (this is illustrated in Case IV of this section).

Source-Oriented Simple-Node Versus Conventional Simple-Node. We now examine the performance of source-oriented and conventional simple-node schemes under two QoS parameters. We use CSPA to construct the full-mesh, and then perform TA. The performance of these two schemes is shown in Figure 15. The source-oriented simple-node significantly improves the SR over the conventional one. For the sparse topology, both schemes have zero CR. When the topology is dense, the CR of the source-oriented approach is nonzero but is still sufficiently small (less than 10^{-3}, on average). Similar trends were observed when the logical links used in computing the aggregated topologies are determined based on MPPWCA and MPPBCA.

Source-Oriented Star Versus Conventional Star. The performance of the conventional and source-oriented stars is shown in Figure 16. Not only does the source-oriented star improves the SR over the conventional approach, but it also significantly decreases the CR.

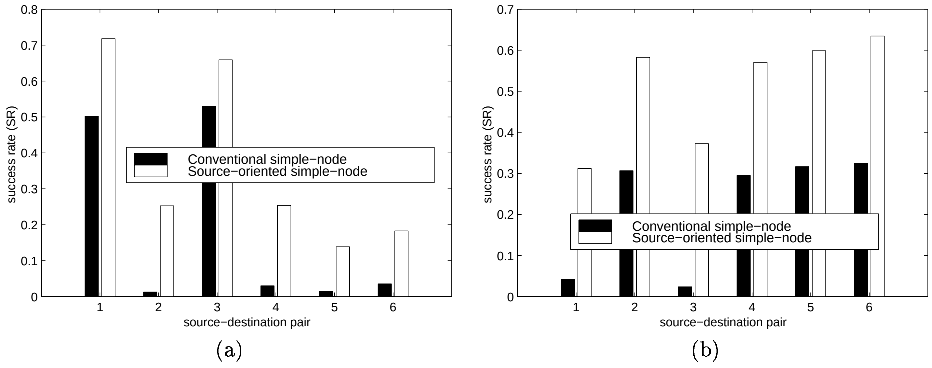
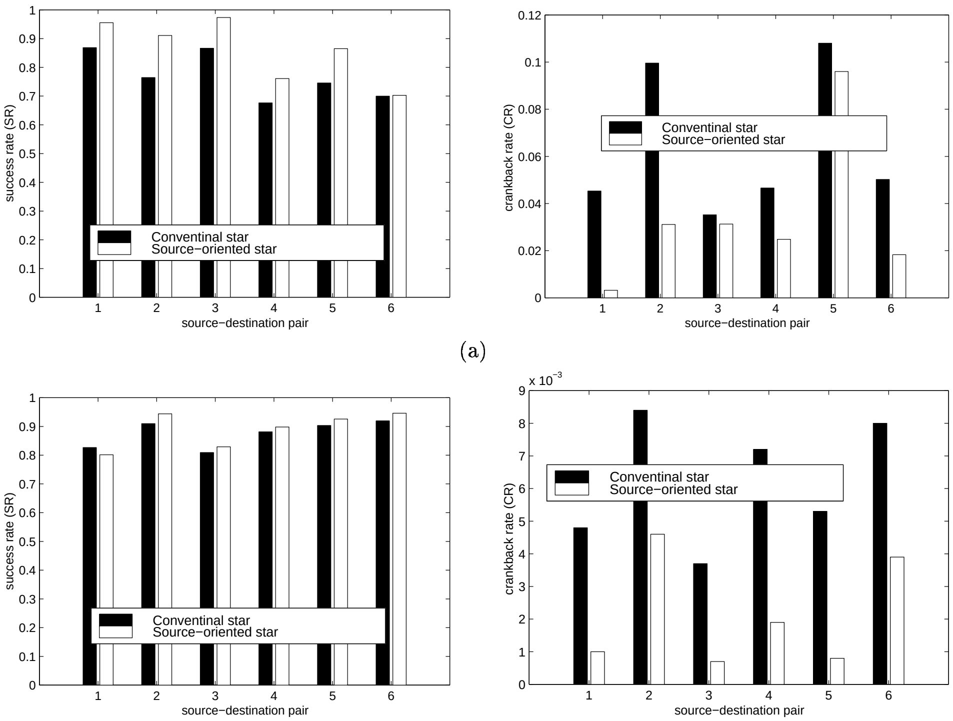


Fig. 15. Performance of two simple-node schemes under two QoS parameters and a static environment: (a) first topology, (b) second topology.

Case III: Three QoS Parameters — Static Environment. We now consider three QoS parameters per link (bandwidth, delay, delay-jitter). Bandwidth and delay values are randomly chosen as in the previous case. The delay-jitter is also randomly chosen from uniform distribution over the interval [10,50(ms)]. The bandwidth and delay requirements of a connection request are randomly chosen as in the previous case. The jitter requirement is uniformly distributed over [100,200(ms)] in the first topology and [150,400(ms)] in the second topology. We use the same source-destination pairs of Table 3.

Finding Logical Links. We first compare CSPA and Modified-MPPBCA along with the conventional MPPBCA and MPPWCA. The performance of these approaches is shown in Figure 17. The modified-MPPBCA with aggressive estimation of s_factor and the conventional MPPBCA schemes have almost the same SR while the modified-MPPBCA slightly reduces CR. Again the modified-MPPBCA with conservative estimation of s_factor reduces CR further at the expense of reducing SR. In terms of the SR, CSPA displays the same trend as in the previous case, i.e., its SR performance is close to that of MPPBCA when the topology is dense and slightly worse than that when the topology is sparse. However, in contrast to MPPBCA, CSPA does not cause any crankback. The SR of MPPWCA is significantly less than the SRs of other approaches, but it has a zero CR and, more importantly, it is less sensitive to network dynamics.

Source-Oriented Simple-Node Versus Conventional Simple-Node. We now examine the performance of source-oriented and conventional simple-node schemes under three QoS parameters. Before performing TA, the full-mesh is obtained using CSPA. The performance of both schemes is shown in Figure 18. The source-oriented simple-node significantly improves the SR over the conventional one. For the sparse topology, both schemes have zero CR. When the topology is dense, the CR of the source-oriented approach (not shown) is nonzero, but it is still sufficiently small (less than 10^{-3} on average).



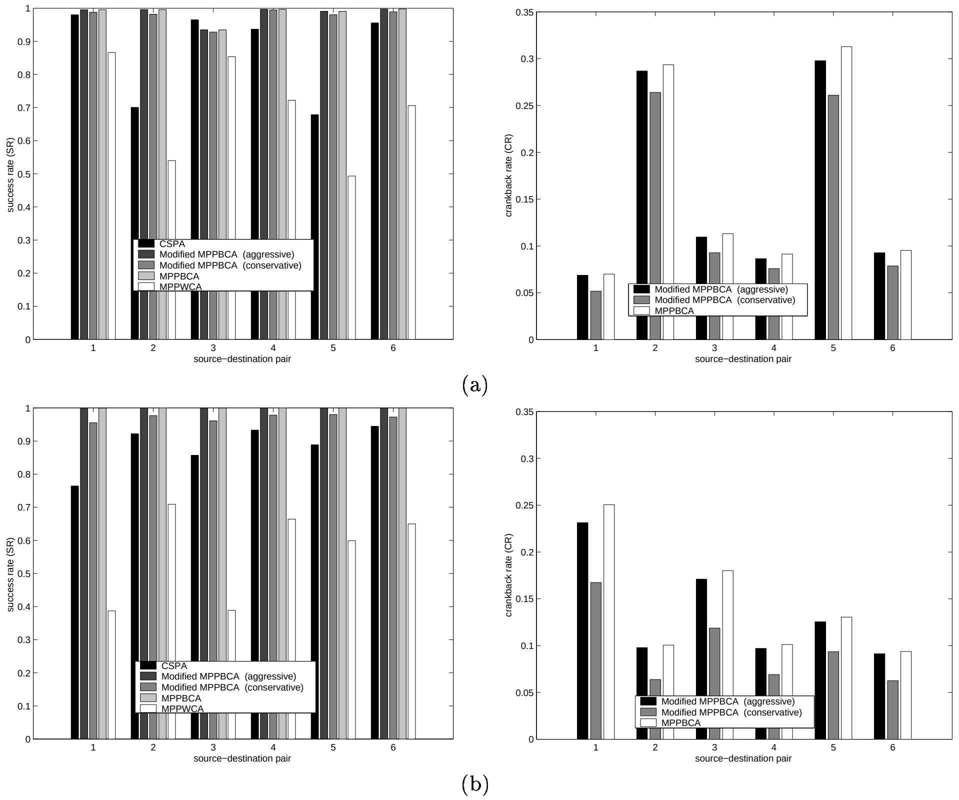
(a)

(b)

Fig. 16. Performance of two star schemes under two QoS parameters: (a) first topology (b) second topology.

Source-Oriented Star Versus Conventional Star. The performance of the conventional and source-oriented stars under three QoS constraints is shown in Figure 19. The source-oriented star improves both the SR and CR over the conventional approach.

Case IV: Two QoS Parameters — Dynamic Environment. In this section, we compare the performance of various TA schemes when network dynamics are taken into account. This includes updating the available link bandwidth after a connection is established and terminated. In addition, we account for the frequency at which TA is performed and state information is advertised. These tasks are done periodically or when triggered by significant changes. In the following simulations, we assume a periodic update policy with different values for the length of the update interval. The available bandwidth and delay over each link are randomly chosen from uniform distributions over the intervals [1,10(Mbps)] and [5,45(ms)], respectively. The bandwidth requirement of a connection request is uniformly distributed over the interval [0.1,10(Mbps)]. The delay requirement is uniformly distributed over the interval [150,200(ms)]. Call holding times are exponentially distributed with mean of 5 minutes. Call inter-arrival times are exponentially distributed with mean of 1 second. Since the same trend was observed under both examined topologies, we only report the results for the first (dense) topology. The source-destination



(a)

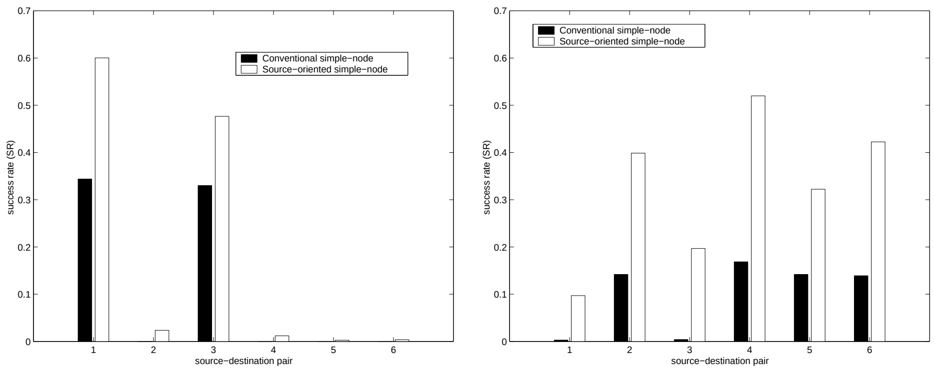
(b)

Fig. 17. Performance of various approaches for computing the weights of the logical links under three QoS parameters: (a) first topology, (b) second topology.

pairs are randomly chosen from nodes 1, 2, 3 of the first topology.

Finding Logical Links. To compare CSPA and Modified-MPPBCA with MPPBCA and MPPWCA in a dynamic environment, we again assume that logical links are determined according to these approaches and all of these links (i.e., the full-mesh) are advertised in every update interval. The performance is depicted in Figure 20 as a function of the update interval. As in the static case, the modified-MPPBCA with aggressive estimation of s_factor and conventional MPPBCA have almost the same SR, but the former scheme has a slightly lower CR. CSPA has a slightly lower SR than MPPBCA and modified-MPPBCA, but it also has a lower CR. The SR of MPPWCA is significantly less than the SR of other approaches, but this scheme has the smallest CR. For all schemes, the SR decreases and the CR increases when increasing the database update interval. Among all schemes, the performance of the MPPWCA is the least sensitive to the length of the update interval.

Source-Oriented Simple-Node Versus Conventional Simple-Node. The performance of the conventional and source-oriented simple-nodes is shown in Figure 21. In both schemes, the value of the update interval has almost no effect on the SR. The CR



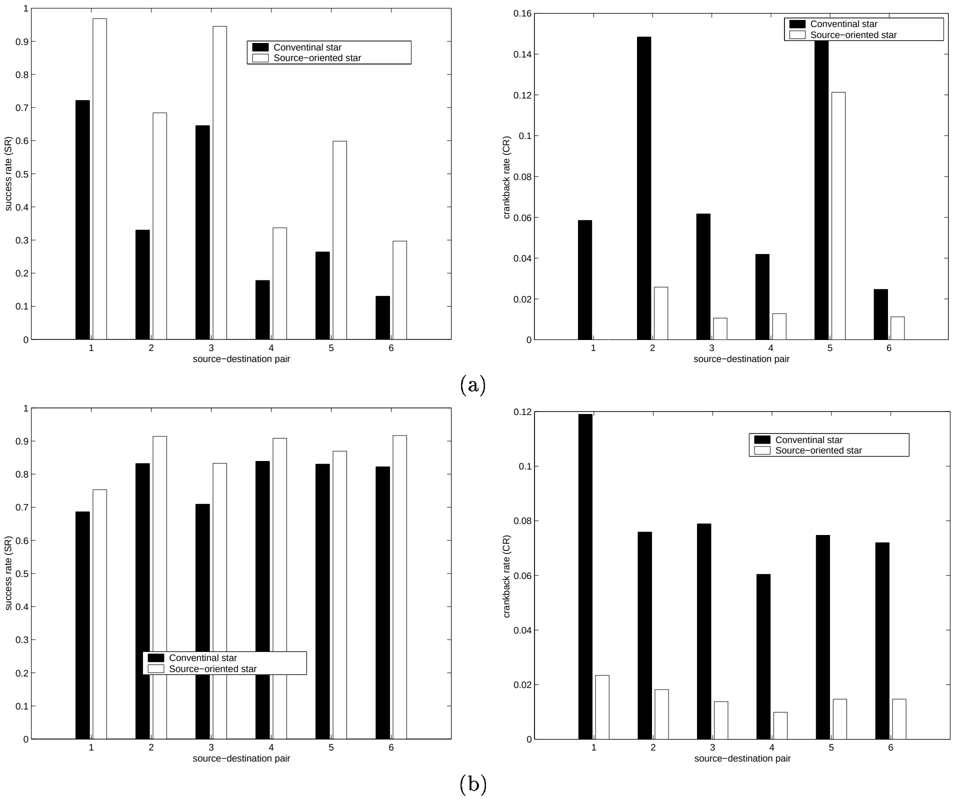
(a) (b)

Fig. 18. Performance of two simple-node schemes under three QoS parameters and a static environment: (a) first topology, (b) second topology.

slightly increases with the update interval. In contrast, in the full-mesh both the SR and CR are sensitive to the value of the update interval (see Figure 20). The SR of the source-oriented simple-node is about three times that of the conventional simple-node at the expense of a slight increase in the CR.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we introduced a source-oriented TA approach for efficient QoS-based routing in scalable networks. The goal of this approach is to eliminate the redundancy in the advertised state information by taking into consideration the relevance of this information for path selection. The source-oriented approach was integrated into three TA schemes (unified quasi-stars, source-oriented simple-node, and source-oriented star), which provide different tradeoffs between compaction and accuracy. The unified quasi-stars scheme is a viable alternative to the conventional full-mesh scheme. It offers the same accuracy as the full-mesh but with less advertisement overhead when the underlying topology is sparse. The other two schemes are more appropriate for dense topologies. For computing the weights of the logical links, we introduced two new approaches (CSPA and modified-MPPBCA), which constitute a compromise between the conventional SPPA and MPPBCA approaches. We studied the performance of our TA schemes via extensive simulations that were carried out for sparse and dense network topologies under static and dynamic scenarios. In the static scenario, we observed that the modified-MPPBCA achieves almost the same SR as the conventional MPPBCA but with lower CR. CSPA performs as good as MPPBCA without causing any crankback. The source-oriented versions of the simple-node and star schemes perform better than their conventional counterparts. We then compared TA schemes using more realistic simulation setup in which the network dynamics (e.g., periodic update of aggregated information, resource allocation following successful call admissions) are taken into account. In this case, we observed that for all schemes the SR decreases and the CR increases with the length of the update interval and that the performance trends are similar to those observed in the static scenario. Simulation results also showed that increasing the



(a)

(b)

Fig. 19. Performance of two star schemes under three QoS parameters: (a) first topology (b) second topology.

update interval has more negative impacts on accurate TA schemes (e.g., full-mesh) than on lossy schemes (e.g., simple node). This is attributed to the fact an accurate state advertisement gradually loses its value as the update interval increases. In a future work, we plan to investigate appropriate periodic and triggered-based updating mechanisms that account for the amount of spatial aggregation.

7. ACKNOWLEDGMENT

The authors would like to thank Whay C. Lee for his constructive comments on an earlier draft of this paper. This work was supported by the National Science Foundation (NSF) under grants ANI 9733143 and CCR-9979310.

REFERENCES

Ahuja, R. K., Magnanti, T. L., and Orlin, J. B. 1993. *Network Flows: Theory, Algorithms, and Applications.* Prentice Hall, Inc.

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

Althofer, I. et al. 1993. On sparse spanners of weighted graphs. *Discrete & Computational Geometry 9*, 1, 81–100.

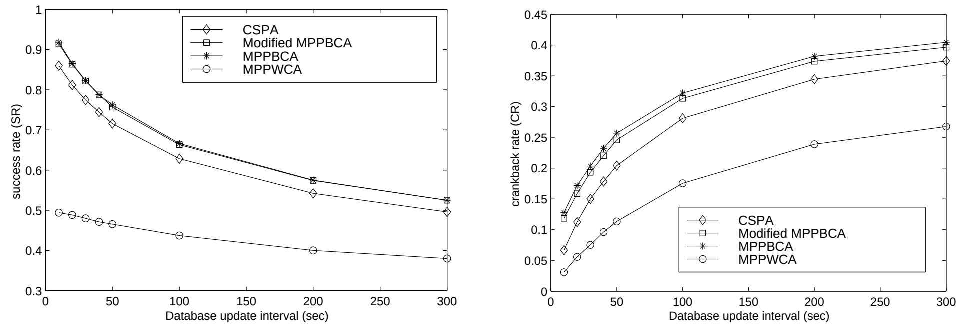


Fig. 20. Performance of various approaches for computing the logical links (dynamic environment with two QoS parameters).

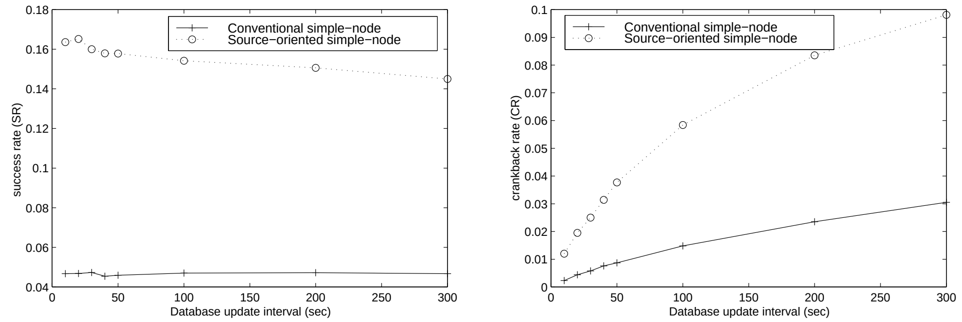


Fig. 21. Performance of two simple-node schemes in a dynamic environment.

Apostolopoulos, G. et al. 1998. QoS routing mechanisms and OSPF extensions. Technical Report draft-guerin-qos-routing-ospf-05.txt (April), Internet Engineering Task Force.

Apostolopoulos, G., Guerin, R., Kamat, S., and Tripathi, S. K. 1998. Quality of service based routing: A performance perspective. In *Proceedings of the ACM SIGCOMM '98 Conference* (Vancouver, British Columbia, Canada, August-September 1998), pp. 17–28.

Awerbuch, B., Du, Y., B., K., and Shavitt, Y. 1998. Routing through teranode networks with topology aggregation. *Journal of High Speed Networks 7*, 1, 57–73.

Awerbuch, B., Du, Y., and Shavitt, Y. 1998. STARS: A simulator for performance study of aggregation based hierarchical routing. In *SCS/IEEE SPECTS'98* (Reno, NV, USA, July 1998).

Awerbuch, B. and Shavitt, Y. 1998. Topology aggregation for directed graph. In *Proceedings of ISCC '98* (1998), pp. 47–52. IEEE.

Basturk, E. and Stirpe, P. 1998. A hybrid spanning tree algorithm for efficient topology distribution in PNNI. In *Proceedings of the 1st IEEE International Conference on ATM (ICATM '98)* (1998), pp. 385–394.

Behrens, J. and Garcia-Luna-Aceves, J. J. 1998. Hierarchical routing using link vectors. In *Proceedings of the INFOCOM '98 Conference*, Volume 2 (March-April 1998), pp. 702–710. IEEE.

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

CLARK, D. ET AL. 1996. Strategic directions in networks and telecommunications. *ACM Computing Surveys 28*, 4, 579–690.

CORMEN, T. H., LEISERSON, C. E., AND RIVEST, R. L. 1996. *Introduction to Algorithms* (Sixteenth ed.). The MIT press and McGraw-Hill book company.

FORUM, T. A. 1996. Private network-to-network interface specification version 1.0 (PNNI 1.0). af-pnni-0055.000.

GARCIA-LUNA-ACEVES, J. J. AND BEHRENS, J. 1995. Distributed, scalable routing based on vectors of link states. *IEEE Journal on Selected Areas in Communications 13*, 8 (Oct.), 1383–1395.

GUERIN, R. AND ORDA, A. 1999. QoS routing in networks with inaccurate information: Theory and algorithms. *IEEE/ACM Transactions on Networking 7*, 3 (June), 350 –364.

GUO, L. AND MATTA, I. 1998. On state aggregation for scalable QoS routing. In *Proceedings of the ATM Workshop '98* (May 1998), pp. 306–314. IEEE.

HAO, F. AND ZEGURA, E. W. 2000. On scalable QoS routing: Performance evaluation of topology aggregation. In *Proceedings of the INFOCOM 2000 Conference*, Volume 1 (2000), pp. 147 –156. IEEE.

HAO, F., ZEGURA, E. W., AND BHATT, S. 1998. Performance of the PNNI protocol in large networks. In *Proceedings of ATM Workshop* (1998), pp. 315 – 323. IEEE.

IWATA, A., SUZUKI, H., IZMAILOV, R., AND SENGUPTA, B. 1998. QOS aggregation algorithms in hierarchical ATM networks. In *Proceedings of the IEEE ICC '98 Conference*, Volume 1 (1998), pp. 243 –248.

JAFFE, J. M. 1984. Algorithms for finding paths with multiple constraints. *Networks 14*, 95–116.

LEE, W. C. 1995a. Spanning tree method for link state aggregation in large communication networks. In *Proceedings of the INFOCOM '95 Conference* (1995), pp. 297–302. IEEE.

LEE, W. C. 1995b. Topology aggregation for hierarchical routing in ATM networks. In *ACM SIGCOMM '95, Computer Communications Review* (1995), pp. 82–92.

LEE, W. C. 1999. Minimum equivalent subspanner algorithms for topology aggregation in ATM networks. In *The 2nd International Conference on ATM (ICATM '99)* (1999), pp. 351 –359. IEEE.

LORENZ, D. H. AND ORDA, A. 1998. QoS routing in networks with uncertain parameters. *IEEE/ACM Transactions on Networking 6*, 6 (Dec.), 768 –778.

MA, Q. AND STEENKISTE, P. 1997. On path selection for traffic with bandwidth guarantees. In *Proceedings of the IEEE International Conference on Network Protocols (ICNP '97)* (1997), pp. 191 –202.

MOY, J. 1998. OSPF version 2. Standards Track RFC 2328 (April), Internet Engineering Task Force.

RAGOZINI, A. R., ROUGIER, J. L., GRAVEY, A., AND KOFMAN, D. 1999. Analysis of the performance of a hierarchical PNNI network. In *The 2nd International Conference on ATM (ICATM '99)* (1999), pp. 365 – 374. IEEE.

ROUGIER, J. L., KOFMAN, D., AND GRAVEY, A. 2000. Optimization of hierarchical routing protocols. *Performance Evaluation 41*, 2-3, 227–245.

TSAI, W. T., V., R. C., TSAI, W. K., AND NISHIGUCHI, O. 1989. An adaptive hierarchical routing protocol. *IEEE Transactions on Computers 38*, 8 (August), 1059–1075.

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

VAN MIEGHEM, P. 1997. Estimation of an optimal PNNI topology. In *Proceedings of the IEEE ATM Workshop* (25-28 May 1997), pp. 570–577. IEEE.

WANG, Z. AND CROWCROFT, J. 1995. Bandwidth-delay based routing algorithms. In *Proceedings of the GLOBECOM '95 Conference*, Volume 3 (Nov. 1995), pp. 2129–2133. IEEE.

WANG, Z. AND CROWCROFT, J. 1996. Quality-of-service routing for supporting multimedia applications. *IEEE Journal on Selected Areas in Communications 14*, 7 (September), 1228–1234.