On Scalable QoS Routing: Performance Evaluation of Topology Aggregation

Fang Hao Ellen W. Zegura
Networking and Telecommunications Group
College of Computing
Georgia Tech, Atlanta, GA 30332
{fang,ewz}@cc.gatech.edu

Abstract

A number of important questions remain concerning the scalability of networks with quality of service guarantees. We consider one of these questions: can QoS routing scale to large networks? To address this question, we evaluate performance of four QoS routing algorithms both with and without topology aggregation, based on simulations of relatively large, structured networks. Among our observations, we find — contrary to intuition — that topology aggregation does not always have a negative impact on routing performance. Aggregation can reduce the routing information fluctuation, increase stability, and thus benefit routing performance. We also propose two new methods of aggregating routing information. Our hybrid aggregation method has performance much better than conventional star aggregation and approaches unaggregated performance. Our weighted aggregation method, while intuitively appealing, offers mixed performance across topologies.

Keywords: quality of service, routing, topology aggregation

1 Introduction

One barrier to large-scale QoS deployment is the scalability of QoS routing. Techniques that enhance the scalability of QoS routing (e.g., aggregation and less frequent routing advertisements) are typically accompanied by a degradation in performance (i.e., more connections that cannot be routed). To complicate matters, routing algorithms also contribute to performance and may interact with scalability techniques in complex ways.

Considerable previous work has considered the performance of QoS routing algorithms [14, 20, 12, 21, 7], but typically without addressing scalability issues. For instance, Ma et al. [12] compare the performance of difference routing algorithms, focusing on the effect of traffic load. To achieve scalability, researchers have proposed various ways to reduce QoS routing overhead due to frequent routing updates [1] or path computations [19, 1, 2]. An alternative technique, namely *topology aggregation*, is one of the most important ways to reduce routing overhead, since it may reduce the amount of update information by orders of magnitude. Lee [11] provides some guidelines for topology aggregation in ATM PNNI networks, and suggests aggregation methods such as symmetric star, full-mesh, spanning tree and complex node representation. Others have also presented various aggregation techniques [15, 10]. However, a systematic evaluation of the performance of different routing algorithms under topology aggregation is lacking in previous work.

In this paper, we examine methods for reducing the overhead associated with QoS routing information. As a first step towards scalability, we assume that the network is organized into domains, and that only aggregated routing information is advertised outside a domain. We focus on decreasing the quantity of routing information exchanged between routing domains, using topology aggregation. Using a detailed hierarchical QoS routing simulator, we evaluate the performance of five routing algorithms on relatively large topologies (e.g., 56 and 200 routers).

We find that, contrary to intuition, topology aggregation does not always have a negative impact on routing performance. Aggregation can reduce the routing information fluctuation, increase stability, and thus benefit routing performance. We also observe that among the four QoS routing algorithms we evaluate, widest-shortest path routing is the most insensitive to change in routing update intervals. Shortest distance routing has the best performance under short routing update intervals. Under aggregation, performance of shortest distance routing and competitive call routing are less predictable than widest-shortest path routing, due to the characteristics of their cost functions.

We also propose two new methods of aggregating routing information. Our *hybrid aggregation method* for widest-shortest path routing advertises hop count information less frequently and in more detail, while advertising available bandwidth more frequently and in less detail. We show that the performance is much better than conventional star aggregation and approaches unaggregated performance. Our *weighted aggregation method* takes into account the frequency of use of paths through a domain. The idea is to make routing information that is used more frequently more accurate. Though intuitively appealing, we find that this method offers mixed performance across topologies.

In the next section, we first give an overview of the QoS routing architecture that our study is based on, with focus on route selection algorithms and topology aggregation approaches. In Section 3, we introduce the configuration of our simulation, including the simulation software, network topology and traffic model. We then evaluate the performance of the routing algorithms for both with and without topology aggregation using simulation in Section 4. In Section 5, we propose two new aggregation approaches based on our observations from evaluation, Finally, we conclude in Section 6.

2 Overview

Researchers have proposed various QoS routing architectures [3, 6, 5, 8]. Our simulation study is mainly based on the PNNI phase one [3] specification, since it is currently the only standardized QoS routing protocol. But we try to generalize our assumptions on the architecture whenever possible. Hence we believe the results should be useful as a reference for performance study of other QoS routing architectures as well. In the following, we first introduce our basic assumptions on the routing architecture, then focus on the routing algorithms and topology aggregation methods.

2.1 Routing architecture

We assume the following QoS routing architecture: Network state information such as available bandwidth on a link is periodically updated throughout the network, so that each router can maintain its own routing table based on the updated information. Connections are routed based on their QoS requirements and the network state information maintained at the routers. In order to achieve scalability, we further assume that the network has a hierarchical structure, similar to the

Internet. Routers in the network are grouped into domains. Only aggregated routing information is advertised outside a domain. Thus a router has detailed information about its own domain and aggregated information about all other domains. Note that we can build a multi-level routing hierarchy by grouping a number of neighboring domains together to form higher level domains, but we focus on the two-level hierarchy in this study.

We also assume source routing is used in the network: the routing path is selected at the router where the connection starts. If the connection starts and ends in different domains, only a "skeleton" path is constructed initially, which specifies some intermediate domains on the path. Detailed path in each intermediate domain is filled in when the connection setup request enters the domain.

The QoS requirements of connections may include bandwidth, delay and jitter. We focus on routing of connections with bandwidth guarantees in this study.

2.2 Route selection methods

We consider five route selection methods: widest shortest path [8], shortest widest path [20], competitive call routing [7], shortest distance [13], and static minimum hop routing. The first four methods are QoS sensitive. Static minimum hop routing does not depend on any QoS metrics; we use it as a baseline for performance comparisons. A brief summary of each routing algorithm is given in Appendix A.

A route selection algorithm usually considers two goals: maximizing efficiency of bandwidth utilization (i.e., choose short paths) and balancing traffic load (i.e., avoiding congested links). The algorithms can be classified according to the way they trade-off between the two goals.

Static minimum hop routing only considers the first goal: it always selects the minimum hop path in a non-greedy way. It may reject a connection because no minimum hop paths are feasible, even though a longer path may exist. Widest-shortest path routing considers both goals, but gives priority to minimizing path length: it only tries to balance load when multiple minimum hop paths are available. Also, it selects a path greedily, in that it considers longer paths if the minimum hop path is not available. Shortest-widest path routing also considers both goals, but gives priority to balancing load. It only tries to select the shortest path when there are multiple paths with the same available bandwidth.

Both competitive call and shortest distance routing try to dynamically trade-off between the two goals. In both algorithms, link cost is defined as a function of available bandwidth; path cost is defined as the sum of link costs. Furthermore, in both algorithms, when the link is lightly utilized, the link cost changes slowly with changes in available bandwidth; when the link is heavily utilized, the link cost changes quickly with changes in the available bandwidth. Thus both algorithms try to maximize efficiency when no congestion occurs, and try to balance load when some links are congested. The difference between the two algorithms is the "shape" of their link cost functions: competitive call routing has an exponential function, which drops slower with the increase in available bandwidth, compared with the hyperbolic function used in shortest distance routing. Figure 1 shows their cost functions on OC-12 links (622 Mbps).

Note that widest-shortest and shortest-widest routing use two cost metrics to characterize a path: hop count and available bandwidth. Competitive call and shortest distance routing just use one metric. We will show later that the cost metrics affect routing performance significantly.

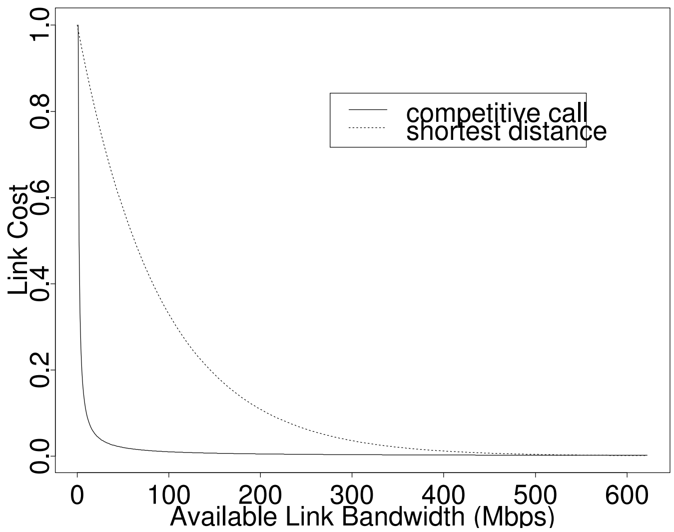

Figure 1: Link cost functions of competitive call and shortest distance routing

2.3 Topology aggregation

Topology aggregation is the key technique for achieving scalability in QoS routing. It is the process of representing the routing information of a domain in a "compact" manner. The aggregated information is used for the network nodes outside this domain to make routing decisions. We call the interface link between the routers inside and outside a domain a *port* of this domain. Thus a routing domain can be considered by the outside world as a single logical node with ports connected to the outside. For a router outside the domain to make routing decisions, the only information it needs to know about this domain are the routing costs between different ports of this domain, which we call *port-to-port distances*.

One way to get the port-to-port distances is to select the "best" path between each port pair, and let the distance between a port pair be the cost of the shortest path between this pair. The definition of "best" depends on the route selection algorithms. For instance, for widest-shortest path routing, the "best" path is the widest-shortest path. Note that, theoretically, the routing algorithms used here to select the best path between port pairs of a domain could be different from that used by the routers outside this domain in routing connections, as long as the aggregated information contains the routing metric that the outside routers need. For example, we could use static minimum hop routing in aggregation, and let port-to-port distances be the costs of the minimum hop path between port pairs. As the same time, we may use shortest distance routing at each router to select path for each connection. However, using a uniform routing algorithm across the network seems to make more sense from a performance point of view. We assume the same routing algorithm is used both for aggregation and routing of connections in our study.

A large domain usually has many ports. How much scalability is achieved by aggregation depends mostly on how the port-to-port distances are represented. Here we consider the two most common ways to represent the port-to-port distances: full mesh and symmetric star. In the full-mesh representation, we represent the port-to-port distances by a matrix, with one entry per pair of ports. Thus a domain with n ports has a mesh of size n^2. The only information that we lose in this aggregation is the correlation between different port pair distances, since the path between different port pairs may share common links. However, this information is not used by any route selection algorithms that we investigate in this study. Thus there is no difference between flat non-hierarchical routing and full-mesh aggregation in terms of routing performance, so we do not distinguish them further [1] . Thus full-mesh representation is very accurate, but does not scale well with the increase in domain size.

In the symmetric star representation, we assume that the topology of the domain is symmetric, i.e., the distances are the same between any two ports. Thus we represent the distances by a single parameter, the *radius* of the domain. Symmetric star representation is much more scalable than full-mesh, since it reduces the routing information size complexity into $O(1)$. The price it pays is the loss of routing information accuracy.

There are a few alternatives for calculating the radius: using the average, the smallest or the largest of all port-to-port distances, or simply a constant distance such as zero. Constant distance cannot give any information about the cost inside the routing domain; and using the smallest or largest distances seem to be biased. For instance, if the paths between most port pairs are all very congested, but the path between one port pair is very lightly used, taking the smallest distance would essentially tell the outside world that this domain is not congested at all. In this case, the advertised distance value would be too "optimistic". Using the largest distance would have similar problem, since its advertised distance value would be too "pessimistic". Hence using the average value seems to be the most reasonable choice. Note the path cost metric for widest-shortest and shortest-widest path routing is a combination of hop count and available bandwidth, so the average port-to-port distance is defined as the combination of average hop count and available bandwidth of all port-to-port distances.

In the remaining discussion, we do not distinguish full-mesh aggregation and unaggregated routing and call them both *routing without aggregation*; we also use *routing with aggregation* to refer to star aggregation. A key aspect of this paper is a thorough investigation of routing performance both with and without aggregation.

3 Simulation Configuration

Our experiments are done on a virtual PNNI testbed [16, 9] written in the new network description language TeD [17, 18]. We describe the network topology and traffic model in the following.

3.1 Network topologies

We mainly use two different topologies to evaluate the performance of various route selection algorithms under different aggregation methods. The two topologies have 56 and 200 router nodes, respectively. Both topologies are first generated by the GT-ITM graph generation package [2]. This package supports the generation of network topologies to reflect the transit-stub domain structure of the Internet, or any wide-area network comprising multiple administrative domains [4]. The

[1] There are actually some other differences between flat routing and hierarchical routing using full-mesh aggregation. For example, under hierarchical routing, if the source and destination nodes are not in the same routing domain, the source will not know the detailed topology of the destination domain, thus can only select routes based on the position of the domain. Under flat routing, however, the source node knows both the position of the domain and the exact position of destination node in the domain. However, we consider these differences secondary factors for routing performance.

[2] Code available at http://www.cc.gatech.edu/projects/gtitm/

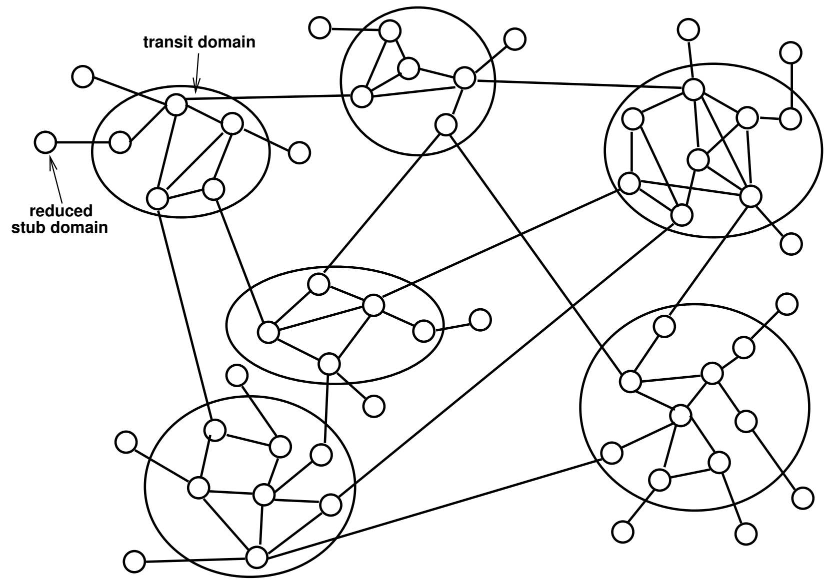

Figure 2: 56 node network topology

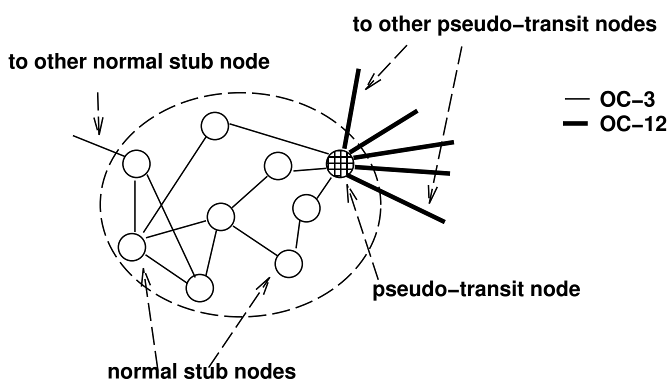

Figure 3: 56 node network topology

routers are organized into two types of domains: transit and stub. Stub domains only carry traffic that originates or terminates internal to the domain. Transit domains provide backbone connectivity to carry traffic that originates in one stub domain and terminates in another. Stub domains may also be connected by direct (stub-stub) edges, used to bypass the backbone for traffic between the connected stub domains.

The 56 node topology is shown in Figure 2. It has six transit domains and 16 stub domains. Each node in the figure represents a router. A host is attached to each router in the stub domains (not shown in the figure). Note that we artificially reduce each stub domain into only one node, for the following two reasons. First, the focus of this study is on routing behavior under different aggregation methods. Aggregation methods affect inter-domain connections more directly than intra-domain connections because aggregated routing state information is only used for routing inter-domain connections. Thus the inter-domain connections, and more particularly, connection rejections that occur during inter-domain routing, are more interesting to our study. By shrinking

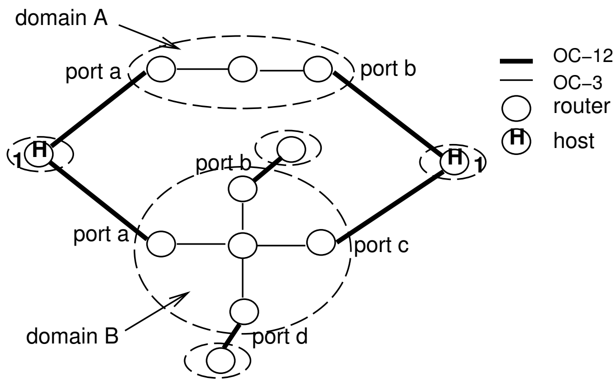

Figure 4: 12 node network topology

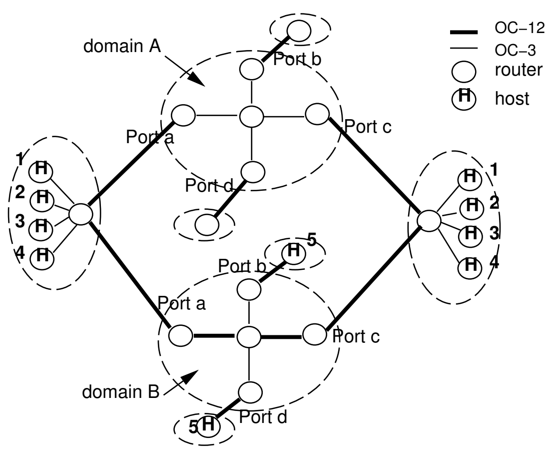

Figure 5: 24 node network topology

Experiment configs	12 node	24 node	56 node 12-12 config	56 node 3-12 config	200 node
Arrival rate (1/s)	0.154	0.25	0.25	0.125	0.028
Holding time (s)	160	120	180	180	160
Max Req. Band. (Mbps)	10	5	10	5	3
Min Req. Band. (Mbps)	1	1	1	1	1

Table 1: Traffic parameters in the experiments

the stub domain size into 1, we essentially eliminate the intra-domain calls. Second, reducing the stub domain size makes simulation much more computationally efficient.

We assign link bandwidth in two ways: (1) all links are OC-12 (622 Mbps), and (2) links inside each routing domain are OC-3 (155 Mbps) but cross-domain links are OC-12 (622 Mbps). We refer to (1) as *12-12 config* and (2) as *3-12 config* in later discussion. We will show later on that the two configurations have very different impact on performance under aggregation.

We also use a 200 node network, which consists of 20 stub domains, with ten routers in each domain on average. The average node degree is 1.93. Again, one host is attached to each stub domain node. There is no transit domain in the network. Instead, each stub domain has a special *pseudo-transit node*. The pseudo-transit nodes of different domains are connected together to provide main inter-domain connectivity. We assign all links between pseudo-transit nodes to be OC-12 and all other links to be OC-3. Figure 3 (b) illustrates the structure of a domain. Note the domain is connected to the outside by many OC-12 links provided by a single pseudo-transit node, and one or two OC-3 links. This asymmetric configuration creates some interesting behavior, as we will show later on.

In addition, we use two other simpler topologies to investigate the effect of the link cost function on aggregation. The topologies consist of 12 and 24 router nodes, as shown in Figures 4 and 5, respectively.

3.2 Traffic model

Connection requests are generated by hosts. We assume connections arrive following a Poisson distribution. Each connection, once admitted, lasts for a period selected from an exponential distribution. The amount of bandwidth requested by each connection is uniformly distributed. The parameters under each experiment configuration is shown in Table 1. We study two traffic patterns: uniform and hot spot traffic. Under uniform traffic, destinations of connections are selected uniformly from all hosts in the network. Under hot spot traffic, destinations are selected with different probabilities: a host may request connection to another particular host with higher probability, and to the rest with lower probability. We set the probability to be 0.8 in our experiments with the 56 node network, and 1.0 in experiments with 12 node and 24 node networks.

4 Performance Evaluation

In this section, we evaluate the performance of the route selection methods under different routing update intervals both with and without aggregation. We adopt bandwidth rejection rate [12] as the major performance metric. Bandwidth rejection rate is defined as the amount of bandwidth of

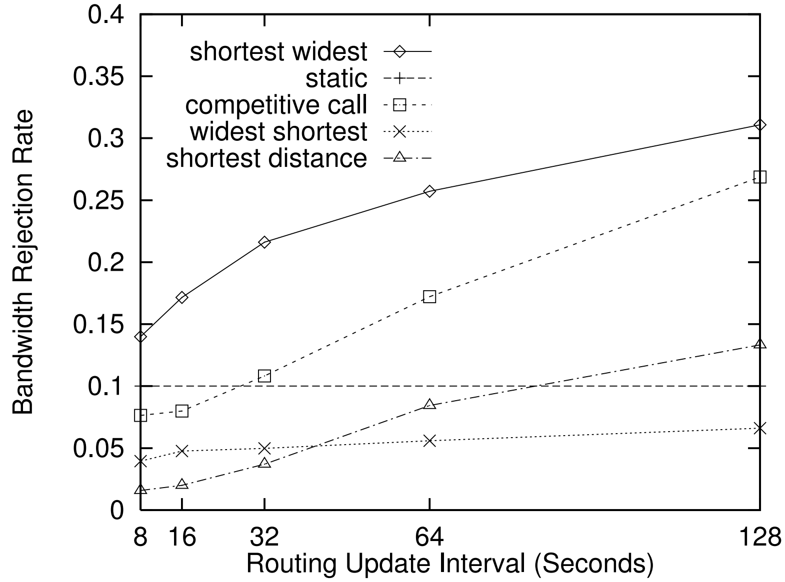

Figure 6: Five routing algorithms under the 56 node network

rejected connections divided by the total amount of bandwidth requested by all connections. We also measure the connection rejection rate, i.e., the *fraction* of rejected connections. But since we find the two metrics are very close in all our experiments, we only show the bandwidth rejection rate.

In the following, we first evaluate the performance of the five routing algorithms without aggregation, comparing their performance under different routing update intervals. Then we investigate their performance with aggregation. We first make general observations on the impact of aggregation on performance of all routing algorithms, and then show more observations on the behavior of shortest distance and competitive call routing under aggregation, as well as the impact of network configuration on aggregation.

4.1 Without aggregation

Figures 6 and 7 shows the bandwidth rejection rate as routing update interval increases under the five routing algorithms. Figure 6 uses the 56 node network with 12-12 config and uniform traffic. Figure 7 uses the 200 node network configuration with uniform traffic. We make two main observations:

Sensitivity to routing update interval. Obviously, performance of static routing does not change with routing update interval since it is solely based on the minimum hop count of paths. Of the four routing algorithms depending on dynamic QoS information, we find widest-shortest path routing is the most insensitive to the change in routing update interval. The other three algorithms (shortest distance, competitive call and shortest-widest path) perform much worse than widest-shortest path routing when the routing update interval is long. The reason is that widest-shortest path routing depends more on hop count information than on available bandwidth information since it considers shortest paths first; while other routing algorithms give relatively more priority to avoiding congested links. Hop counts may change only when a feasible minimum hop path becomes infeasible, or vice versa, which occurs much less frequently than change in

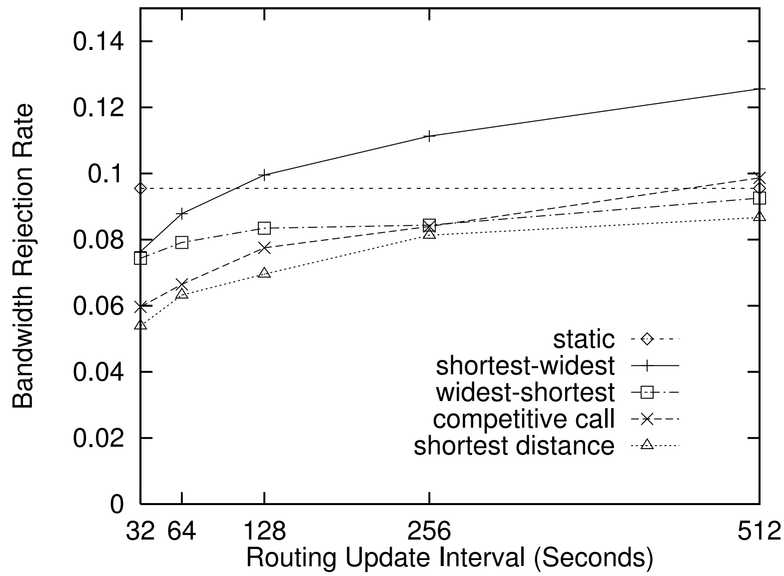

Figure 7: Five routing algorithms under the 200 node network

available bandwidth.

Performance under short update interval. Shortest distance routing has the best performance among all routing algorithms under short routing update intervals. This is because it represents a good trade-off between the two routing goals: it avoids heavily congested links, and selects the path with minimum hop count if multiple paths with roughly the same level of congestion exist. Competitive call routing appears to be worse than shortest path routing, although it tries to do the same thing. The difference is caused by their link cost functions. The link cost of the shortest distance routing is insensitive to the change in available bandwidth over a wider range than competitive call routing. When the links are not very congested, shortest distance routing behaves similar to widest-shortest path routing because adding an extra hop increases the path cost more than having a more congested link. Competitive call routing, on the other hand, gives more priority to congestion avoidance, i.e., load balancing. However, since we have not fully experimented with different parameters of the competitive call routing's cost function, we cannot conclude that competitive on line routing is worse in all cases. But we do observe that it is not straightforward to find good parameter values, a disadvantage compared with shortest-distance routing.

In summary, we find that shortest distance routing has the best performance under short routing updates; widest-shortest path routing is best under long routing updates. Widest-shortest path is the most insensitive to change in the routing update interval, among the four algorithms that depend on QoS information.

4.2 Impact of aggregation

Clearly, aggregation reduces the accuracy of routing state information. Hence intuitively, we should expect to observe a degradation in routing performance under aggregation. However, we find that aggregation does not always have a negative impact on performance. When the routing update interval is long, routing performance suffers from network state instability. Aggregation

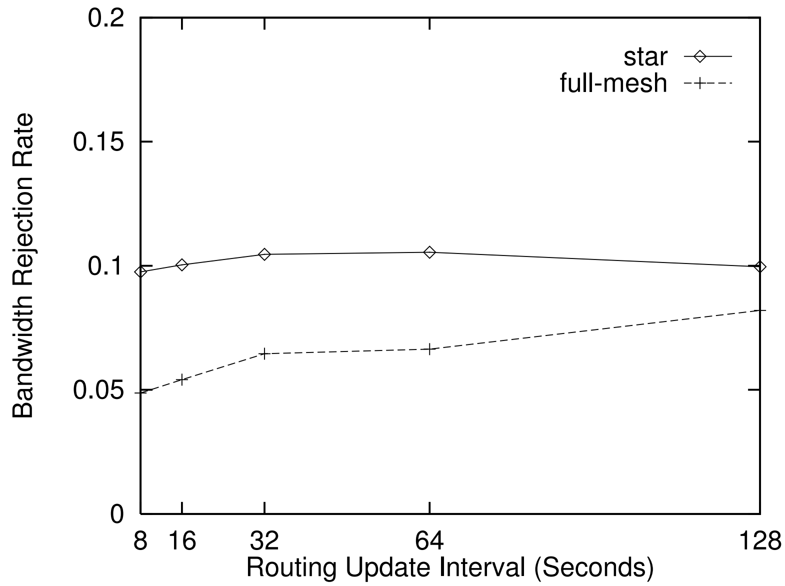

Figure 8: Impact of aggregation: widest-shortest routing

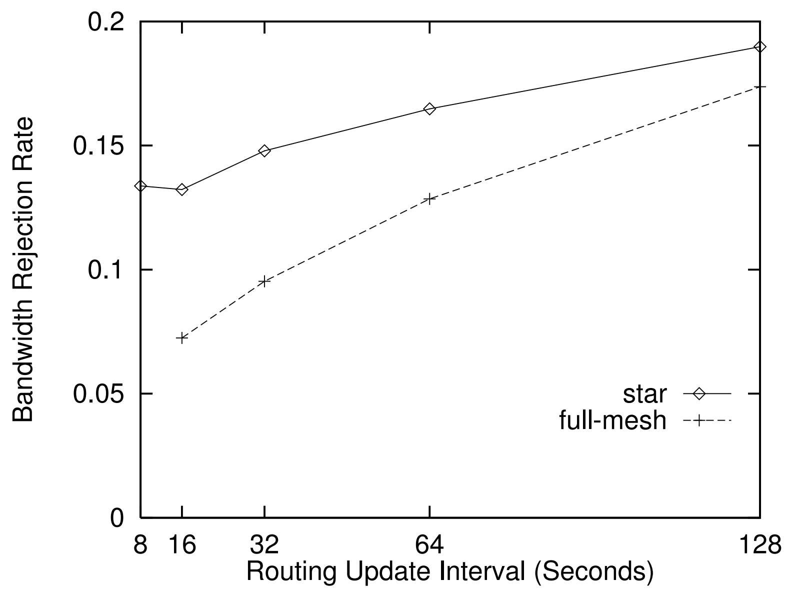

Figure 9: Impact of aggregation: shortest-widest routing

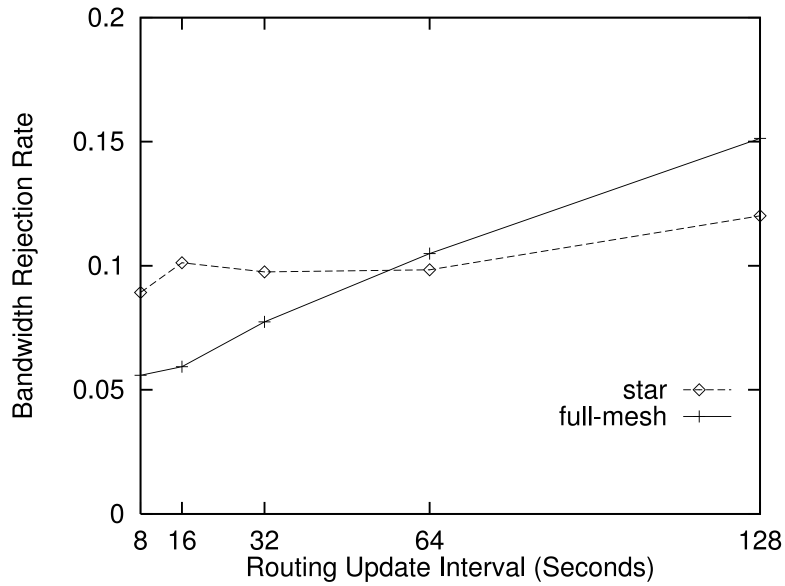

Figure 10: Impact of aggregation: competitive call routing

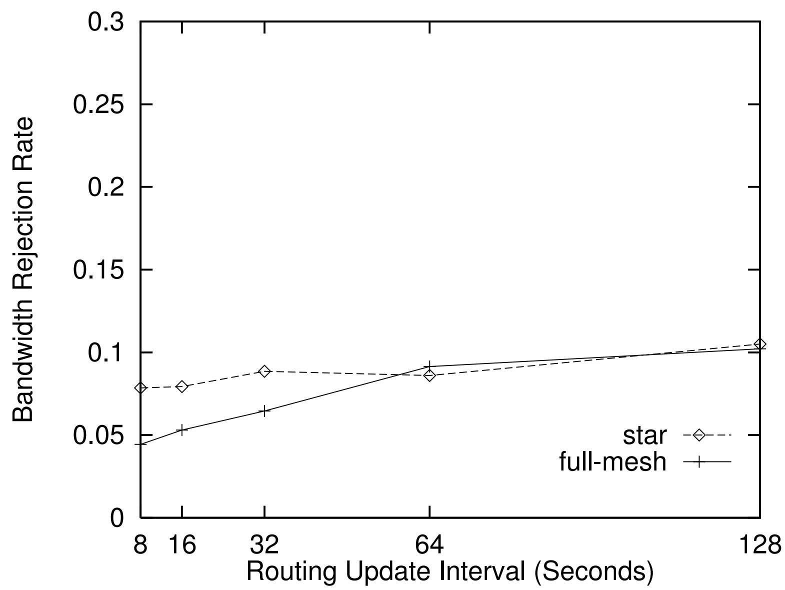

Figure 11: Impact of aggregation: shortest distance routing

	Port pair index			
	1	2	3	4
unaggregated routing	0.032	0.184	0.156	0.379
average of unaggregated routing	0.041	0.116	0.062	0.125
aggregated routing	0.012	0.102	0.088	0.131

Table 2: Standard deviation of sampled port-to-port distances

can reduce the network state fluctuation and make routing more stable. The argument follows the central limit theorem: if we assume the distances between different port pairs of the same domain are independent identically distributed (i.i.d.) random variables, the average of all port-to-port distances of this domain should have smaller variance than the individual distances.

As a result, the performance difference between unaggregated and aggregated routing under long update intervals is smaller than under short update intervals, and sometimes the aggregated routing even performs better than the unaggregated routing. We observe this effect under all the four QoS routing algorithms. Figures 8, 9, 10, and 11 show the performance of widest-shortest, shortest-widest, competitive call, and shortest distance routing with and without aggregation, under the 56 node network with 3-12 config.

To confirm that aggregation reduces the variance of the port-to-port distances, we log the port-to-port distances during simulation of the 56 node network using competitive call routing. Then we randomly select four routing domains, and one port-pair within each routing domain to get the standard deviation of the distances of these port-pairs. We calculate the standard deviation for both simulations with and without aggregation. We also take the data collected from unaggregated routing, and calculate the standard deviation the average of the port-to-port distances for each selected domain. In this way, we hope to show that by averaging the *same* data collected from unaggregated routing, we can indeed reduce the variance. Table 2 shows the standard deviation of the port-to-port distances. We observe that the standard deviation of the data either from aggregated routing or average of unaggregated routing is significantly smaller than unaggregated routing, consistent with our argument. The standard deviation for port-pair 1 is small in all three cases because there is no congestion in that routing domain and thus port-to-port distances do not change much during simulation. Note that typical distance values range from 0 to 1, so the variances of the distances are fairly large.

We need to emphasize here that it is *not always true* that the advantage of unaggregated routing over aggregated routing becomes smaller or diminishes with the increase in routing update intervals. The above observation holds if the port-to-port distances for each domain are roughly i.i.d.. For instance, we have shown that under the 200 node configuration (the same one as we used in this study), the advantage of unaggregated over aggregated routing becomes larger with the increase in routing update intervals [9]. This is because the port-to-port distances are evidently not i.i.d. — the distances between most port pairs are always zero because they are connected with the same pseudo-transit node. The few non-zero distance port pairs cause the variance of the average port-to-port distance under aggregation. Thus for zero-cost port pairs, their variance increases instead of decreases under aggregation.

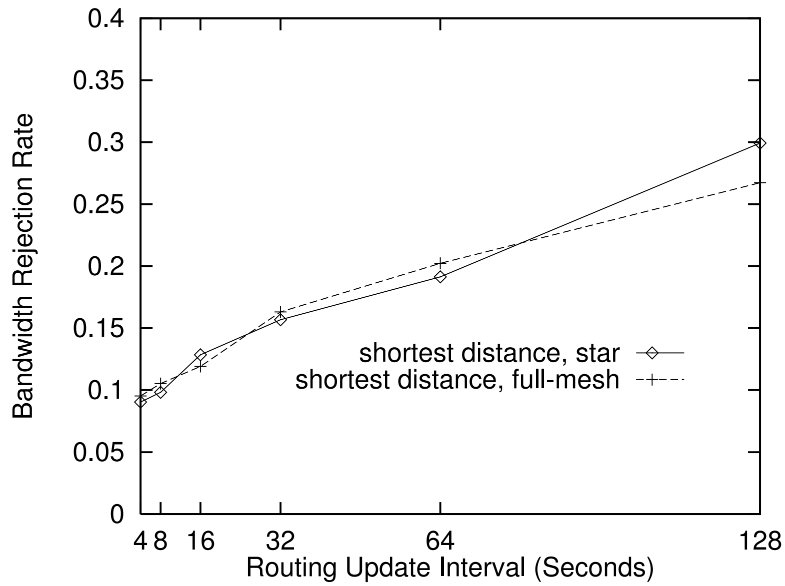

Figure 12: Widest-shortest and shortest distance routing under the 12 node network

4.3 Shortest distance and competitive call routing under aggregation

We make more observations on shortest distance routing and competitive call routing under aggregation in this section. The path cost metrics of both the two routing algorithms are single valued, so hybrid aggregation does not apply here. Furthermore, the link cost metrics of both algorithms are non-linear functions of link available bandwidth. They drop first quickly then slowly with the increase in available bandwidth. We show in this section that their non-linear cost functions affect their routing behavior under star aggregation and make routing performance less predictable. We use shortest distance routing in the following example to illustrate the intuition, but the observation is valid also for competitive call routing.

Consider two port pairs of a routing domain: ab and cd. Assume both port pairs are directly connected by a OC-12 link (622 Mbps). Assume ab is heavily congested and has cost of 1.0 (i.e., only has 1 Mbps bandwidth available); and cd is not congested and has cost of 1/622 (i.e., has 622 Mbps bandwidth available). Intuitively, the aggregated port-to-port distance should reflect an "average state". However, the average distance here is roughly 0.5, which corresponds to only 2 Mbps bandwidth available, very close to the state of ab but far off that of cd. In general, star aggregation under shortest distance routing tends to be accurate for congested port pairs, but inaccurate for uncongested port pairs. It may perform well if most port pairs of a domain are congested, but perform poorly if most port pairs are uncongested. In practice, the congestion state of a domain may change dynamically, thus making it hard to predict the performance under aggregation.

The 12 node topology (Figure 4) shows an example where star aggregation under shortest distance routing performs very well: no traffic passes between port pair bd in domain B but port pair ab is heavily used, so the aggregated cost of domain B should be very accurate for ab. The 24 node topology (Figure 5) shows an example where star aggregation under shortest distance routing does not perform well. Note that port pair bd in domain B should be more congested than ac because the total available capacity on all paths between the source-destination pairs 1, 2, 3

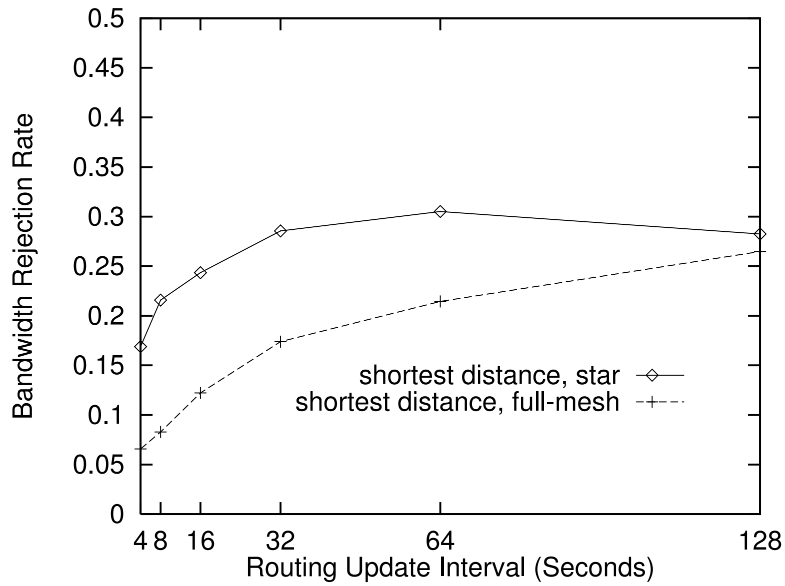

Figure 13: Widest-shortest and shortest distance routing under the 24 node network

and 4 is greater than that between source-destination pair 5. Thus the cost of port pair *ac* is not accurate under aggregation, although most of the traffic passes through it. Performance results from simulation study are consistent with our observation, as shown in Figures 12 and 13.

4.4 Network configuration and aggregation

The impact of aggregation on routing performance is greatly affected by network configurations. For instance, as observed in [9], routing performance are very close for with and without aggregation, if the routing domains are roughly symmetric, but could be significantly different if the domains are asymmetric. Here we show that even under the same domain structure, the effect of aggregation on performance could be very different. We compare performance of widest-shortest path routing under full-mesh and star aggregation by using the same 56 node network but two different configurations: 12-12 config and 3-12 config. Figures 8 and 14 show the bandwidth rejection rate corresponding to 3-12 config and 12-12 config.

The result shows full-mesh and star aggregations perform similarly under 12-12 config but are significantly different under 3-12 config. The reason is: in the first case, most congestion happens at the cross-domain links since all the nodes inside each domain are relatively better connected. Consequently, the accuracy of port-to-port distances of each domain is less important than the cost of the links since routing decisions are mostly based on bottleneck links. However, in the second case, most congestion happens inside each domain since the capacity of inside domain links are much lower than cross-domain links. Thus the accuracy of port-to-port distances of each domain is critical for performance.

Note that in Figure 14, the performance under star aggregation is even a little better than full-mesh aggregation when routing update interval is long. This is consistent with our observation in Section 4.2.

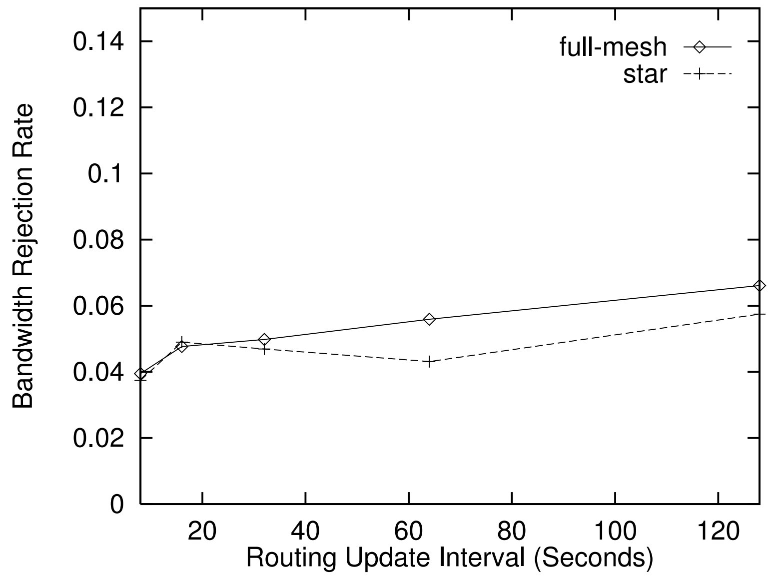

Figure 14: Widest-shortest routing under 56 node network with 12-12 config

5 New Aggregation Approaches

In this section, we propose two new aggregation methods, with the intention of improving the routing performance under aggregation. The first method is applicable only to widest-shortest routing; while the second one could potentially be used for all routing algorithms.

5.1 Hybrid aggregation for widest-shortest path routing

We have shown in the last section that widest-shortest routing is very promising since it performs fairly stable under different routing update intervals. Unfortunately, it is greatly affected by topology aggregation, which limits its scalability. In this section, we propose a new aggregation approach that is a hybrid of full-mesh and star aggregation. The idea comes from the following observation. We know that hop count changes much less frequently than available bandwidth. Thus we may afford to update hop count information less frequently than available bandwidth without much degradation. On the other hand, however, hop count information is more critical for widest-shortest path routing, since the routing decisions are mostly based on hop counts. When we update hop count information more slowly, we can afford to make it more accurate each time.

We take this to an extreme: only advertise hop count information once but advertise it in full-mesh representation. At the same time, we advertise available bandwidth information periodically in star representation. We only advertise hop counts once because even when we switch from a minimum hop path to a longer path, we are likely to switch back to the minimum hop path soon, since the bandwidth might soon be available again. Updating hop count information multiple times on a slow time scale would be likely to cause network state fluctuation.

Figures 15 and 16 show the bandwidth rejection rate for widest-shortest routing under star, corresponding to the 56 node network with hot spot traffic and the 200 node network with uniform traffic, respectively. We observe the bandwidth rejection rate under hybrid routing is very close to full-mesh aggregation and much lower than star aggregation. Thus the hybrid aggregation seems

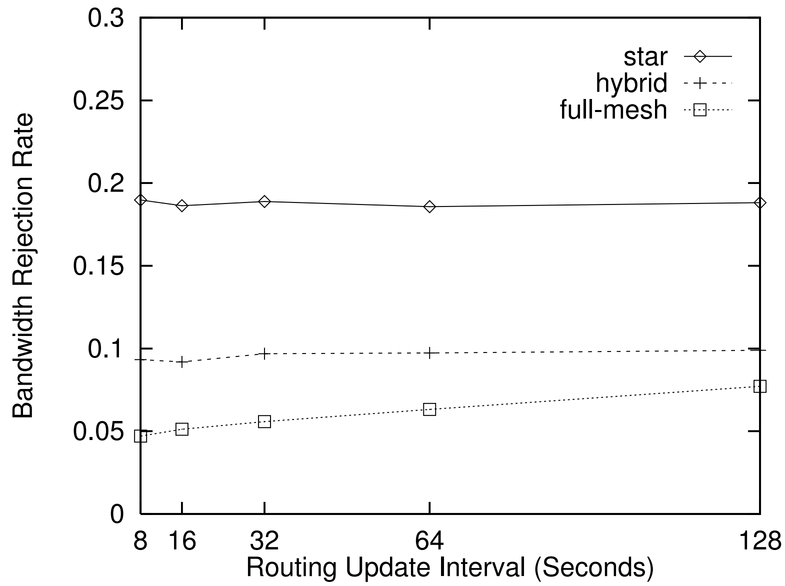

Figure 15: Widest-shortest path routing under aggregation: 56 node network

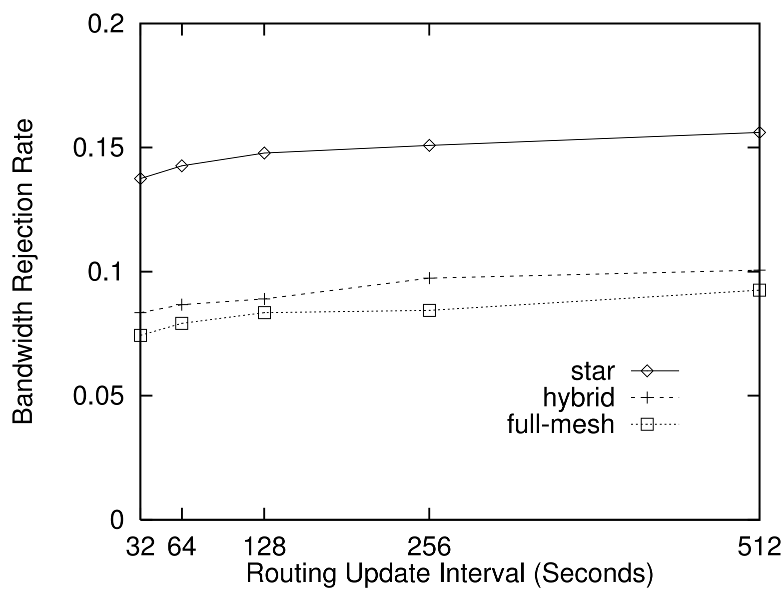

Figure 16: Widest-shortest path routing under aggregation: 200 node network

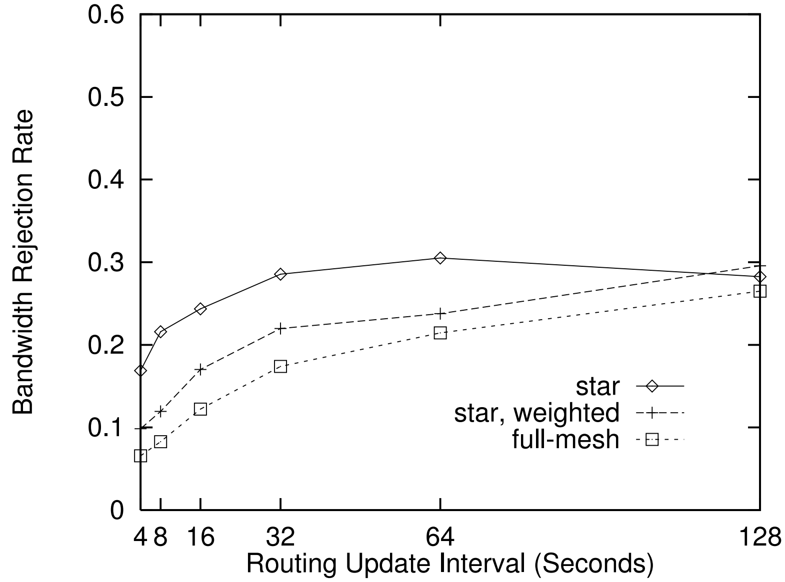

Figure 17: Weighted aggregation under 24 node network

to be a very promising approach for widest-shortest path routing.

5.2 Weighted aggregation

In this section, we propose an alternative version of the star aggregation method, namely, weighted star aggregation. The intuition is as follows. We observe that the accuracy of different port-to-port distances have different impact on routing performance. For instance, in domain A of the 24 node topology, port pair *ac* is more important than other port pairs such as *bd* and *ab*, because *ac* is more heavily used. Hence it seems to make sense to make the distance of *ac* more accurate when we aggregate, with the possibility of sacrificing the accuracy of other port-to-port distances, given the goal of minimizing bandwidth or connection rejection rate.

We achieve this by giving each port pair a weight proportional to the amount of traffic going through it, so that more frequently used port pairs get greater weight. Then we average the port-to-port distances according to their weight. Specifically, assume the distance between port i and j is d_{ij} and the number of connections between i and j is w_{ij}, the aggregated distance will be $\sum_{all\ i,j}(d_{ij} \cdot w_{ij}) / \sum_{all\ i,j} w_{ij}$. The weights are collected by the edge routers periodically. We set the period length to be 100 times longer than the mean connection arrival time, in the hope that it will be long enough to collect enough data points, but short enough to keep up with the change in traffic patterns. Figure 17 shows the bandwidth rejection rate for unweighted star, weighted star and full-mesh aggregation under the 24 node topology. Consistent with our intuition, weighted star aggregation has much better performance than unweighted star, approaching full-mesh.

However, it turns out this idea does not apply equally well to more general networks. Figures 18 and 19 show the performance of weighted and unweighted star aggregation under the 200 node network with uniform traffic and 56 node network with hot spot traffic, correspondingly. Under the 200 node network, weighted star has reasonable performance improvement over unweighted star. But under the 56 node network, the performance of weighted and unweighted star aggregation are very close.

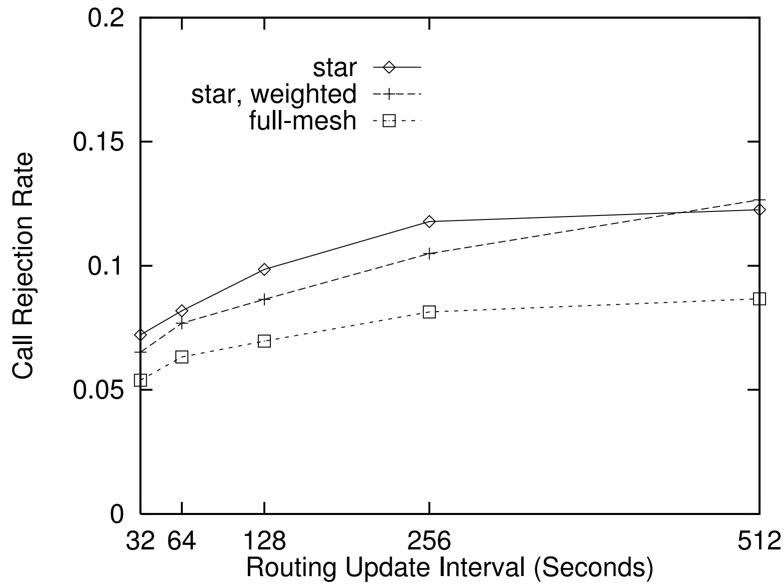

Figure 18: Weighted aggregation under the 200 node network

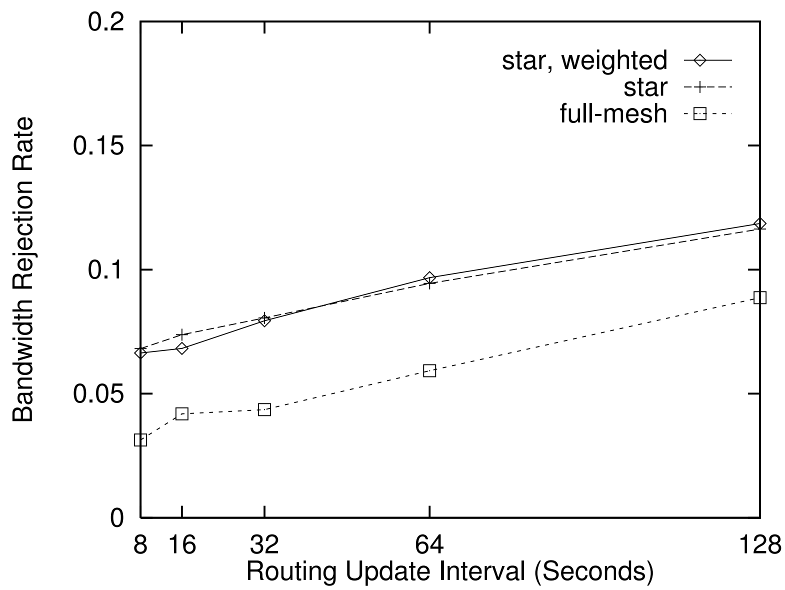

Figure 19: Weighted aggregation under the 56 node network

We suspect the main reason is as follows. The routing decision is made by comparing all available paths between a source and destination pair. Thus a good routing decision depends on the accuracy of the cost of *all* paths between them, instead of just *some* paths. Inaccurate cost of one path may "distract" the routing algorithm and lead to a bad routing decision even when all others are accurate, since the inaccurate path might have the lowest advertised cost while actually the most congested. Similarly, the accuracy of the cost of a path depends on the accuracy of the cost of all the links and domains on the path. Inaccuracy of one domain may cause inaccuracy of the whole path. The weighted aggregation may help only when the traffic between the source and the destination dominates on all paths between them. In that case, the advertised routing cost of all domains is accurate for this pair of source and destination. In the simple 24 node network, this happens to be true: the traffic between source-destination pairs 1, 2, 3, and 4 dominates in both domains A and B, so the cost of both paths between these source and destination pairs is accurate.

Similar behavior occurs in the 200 node network. Remember that in the 200 node network, most port pairs of each domain have the same distance of zero because they are connected with the same pseudo-transit router; they are much more heavily utilized than the few non-zero port pairs. For most source and destination pairs, the paths between them do not go through any unpopular ports of any domain. Thus under weighted aggregation, the routing algorithms are generally not distracted by inaccurate information from unpopular ports.

However, aside from these special cases, accurate costs are usually mixed with inaccurate costs in more general network configurations, therefore the weighted aggregation cannot help, as we observed in Figures 18 and 19.

6 Conclusions

In this paper, we evaluated the performance of four different QoS routing algorithms under different routing update intervals both with and without topology aggregation. We also proposed two new aggregation methods, and compared their performance with conventional star aggregation. Our major conclusions are as follows: (1) Without aggregation, widest-shortest path routing is most insensitive to change in routing update intervals. Shortest distance routing has the best performance under short routing updates; widest-shortest path routing is best under long routing updates. (2) Topology aggregation does not always have a negative impact on routing performance. Aggregation can reduce the routing information fluctuation, increase stability, and thus benefit routing performance. (3) Our hybrid aggregation for widest-shortest path routing has performance much better than conventional star aggregation and approaches unaggregated performance. But our weighted aggregation method offers mixed performance across topologies. Both algorithms have similar scalability as the conventional star aggregation. (4) Performance of shortest distance and competitive call routing under aggregation is less predictable due to their non-linear link cost functions.

A Five route selection algorithms

Widest shortest path. First select all feasible paths with minimum hop counts (shortest); then among them select the one with maximum available bandwidth (widest). If multiple such paths exist, choose one randomly. Path feasibility and bandwidth equality are defined as follows. We set a minimum bandwidth threshold δ to be the connection requested bandwidth. A path is feasible if its available bandwidth is above δ. Similarly, two bandwidth values are considered the same if their difference is no more than δ.

Shortest widest path. First select all paths with maximum available bandwidth (widest); then among them select the one with the smallest hop count (shortest). Choose one randomly if multiple such paths exist. Feasibility and bandwidth equality are defined the same as in widest-shortest path routing.

Competitive call routing. Define the cost metric for link e as $\mu^{\frac{r_e}{b_e}-1}$, where r_e and b_e are occupied and total bandwidth for link e and μ is a parameter. Define the cost of a path as the sum of the link costs. Select the minimum cost path provided it is feasible, i.e., its cost does not exceed a threshold ρ. Selecting appropriate values for parameters μ and ρ turns out to be nontrivial [7, 21]. In this study, we set $\mu = 1000$ and $\rho = 1$ based on experience.

Shortest distance routing. Define cost metric for link e as $\frac{1}{a_e}$, where a_e is the amount of available bandwidth on link e. Define path cost as the sum of link costs. Select the minimum cost path provided it is feasible, i.e., the path cost does not exceed a threshold ρ. We set ρ to be 1 in our experiments.

Static minimum hop. Choose the path with minimum hop count. If multiple such paths exist, choose one randomly.

References

[1] G. Apostolopoulos, R. Guerin, S. Kamat, and S. Tripathi. Quality of service based routing: a performance perspective. In *Proceedings of ACM SIGCOMM'98 Conference*, September 1998.

[2] G. Apostolopoulos and S. Tripathi. On reducing the cost of on-demand QoS path computation. In *Proceedings of the Sixth International Conference on Network Protocols*, October 1998.

[3] ATM Forum. Private Network-Network Interface specification version 1.0. Technical Report af-pnni-0055.000, ATM Forum, March 1996.

[4] K. Calvert, M. Doar, and E. Zegura. Modeling Internet topology. *IEEE Communications Magazine*, June 1997.

[5] I. Castineyra, N. Chiappa, and M. Steenstrup. The nimrod routing architecture. Technical report, RFC 1992, August 1996.

[6] E. Crawley, R. Nair, B. Rajagopalan, and H. Sandick. A framework for QoS-based routing in the Internet. Technical report, RFC 2386, August 1998.

[7] R. Gawlick, A. Kamath, S. Plotkin, and K. Ramakrishnan. Routing and admission control in general topology networks. Technical Report STAN-CS-TR-95-1548, Stanford University, 1995.

[8] R. Guerin, A. Orda, T. Przygienda, and D. Williams. Qos routing mechanisms and OSPF extensions. In *IETF Internet draft* `<draft-guerin-qos-routing-ospf-03.txt>`, January 1998.

[9] F. Hao, E. W. Zegura, and S. Bhatt. Performance of the PNNI protocol in large networks. In *1998 IEEE ATM Workshop Proceedings*, pages 315–323, 1998.

[10] A. Iwata, H. Suzuki, R. Izmailov, and B. Sengupta. QoS aggregation algorithms in hierarchical ATM networks. In *Proceedings of the International Conference on Communications*, 1998.

[11] W. C. Lee. Topology aggregation for hierarchical routing in ATM networks. *ACM Computer Communication Review*, 25(2), April 1995.

[12] Q. Ma and P. Steenkiste. On path selection for traffic with bandwidth guarantees. In *Proceedings of IEEE International Conference on Network Protocols*, Atlanta, GA, October 1997.

[13] Q. Ma, P. Steenkiste, and H. Zhang. Routing high-bandwidth traffic in max-min fair share networks. In *ACM SIGCOMM'96*, pages 206–217, 1996.

[14] I. Matta and A. U. Shankar. Dynamic routing of real-time virtual circuits. In *Proceedings of IEEE International Conference on Network Protocols*, pages 132–139, Columbus, OH, 1996.

[15] M. Montgomery and G. Veciana. Hierarchical source routing through clouds. In *Proceedings of IEEE INFOCOM*, 1998.

[16] K. Perumalla, M. Andrews, and S. Bhatt. A virtual PNNI network testbed. In *Proceedings of the 1997 Winter Simulation Conference*, pages 1057–1064, December 1997.

[17] K. Perumalla and R. Fujimoto. A C^{++} instance of TeD. Technical Report GIT-CC-96-33, College of Computing, Georgia Institute of Technology, 1996.

[18] K. Perumalla, A. Ogielski, and R. Fujimoto. MetaTeD: A meta language for modeling telecommunication networks. Technical Report GIT-CC-96-32, College of Computing, Georgia Institute of Technology, 1996.

[19] A. Shaikh, J. Rexford, and K. Shin. Efficient precomputation of quality-of-service routes. In *Proc. Workshop on Network and Operating Systems Support for Digital Audio and Video*, pages 15–27, July 1998.

[20] Z. Wang and J. Crowcroft. Quality-of-service routing for supporting multimedia application. *IEEE Journal on Selected Areas in Communications*, 14:1228–1234, September 1996.

[21] L. Zhang, M. Andrews, S. Bhatt, and K. Krishnan. A performance comparison of competitive on-line routing and state-dependent routing. In *Proceedings of GLOBECOM 97*, pages 1813–1819, 1997.