

A Comparative Study of QoS Routing Schemes That Tolerate Imprecise State Information

Xin Yuan Wei Zheng Shiling Ding

Department of Computer Science, Florida State University, Tallahassee, FL 32306

{xyuan,zheng,shilding}@cs.fsu.edu

Abstract—

In large networks, maintaining precise global network state information is almost impossible. Many factors, including non-negligible propagation delay, infrequent link state update due to overhead concerns, link state update policy, resource reservation, and hierarchical topology aggregation, have impacts on the precision of the global network state information. To achieve efficient Quality of Service (QoS) routing, a practical routing algorithm must be able to make effective routing decisions in the presence of imprecise global network state information. In this paper, we compare five QoS routing algorithms that were proposed to tolerate imprecise global network state information, safety-based routing, randomized routing, multi-path routing, localized routing, and static multi-path routing. The performance of these routing algorithms are evaluated under two link state update policies, the timer based policy and the threshold based policy. The strengths and limitations of each scheme are identified.

I. INTRODUCTION

To support QoS routing, global network state information is typically maintained by either a distance vector algorithm [8] or a link state algorithm [9]. In this paper, we will assume that a link state algorithm is used to maintain the global network state information. Using the link state algorithm, when a node detects a change of the state of its links, it performs a link state update, that is, it informs the change to all other nodes in the network using a reliable flooding algorithm. The rule to govern when to perform an update is called the *link state update policy*. In large networks, maintaining precise global network state information in the dynamic environment is almost impossible. Many factors, including non-negligible propagation delay, infrequent link state update due to overhead concerns, link state update policy, and hierarchical state aggregation, have impacts on the precision of the global network state information [6].

Depending on the reason that causes the imprecise state information, the nature of the imprecision is different. The imprecision caused by non-negligible propagation delay or infrequent link state update is *random* in the sense that routers do not have sufficient information to determine the actual value of the link state. When the imprecision is caused by link state

update policies, the imprecision is *deterministic* in that routers can usually infer the range of the actual value of the link state and use this information to perform efficient QoS routing. In practice, the imprecision is usually a combination of deterministic imprecision and random imprecision.

Imprecise global network state information can greatly affect the performance of a QoS routing algorithm. It has been shown that a QoS routing algorithm that treats the stale state information as accurate can degrade drastically when the global network state information is imprecise [1], [11]. Hence, mechanisms must be incorporated into a practical QoS routing algorithm to tolerate the imprecise global network state information and make effective routing decisions in the presence of imprecise state information. A number of QoS routing methods that tolerate imprecise state information have been proposed: safety-based routing [2], randomized routing [2], multi-path routing [4] and localized routing [10].

These techniques were proposed for different purposes. It is unclear how effective each technique is in dealing with different types of imprecision and what the relative performance of each technique is. In this paper, we attempt to answer these questions and find methods that can effectively deal with both deterministic and random imprecision through a comparative simulation study of these methods. Notice that although QoS routing, especially multi-path routing, is in general a balance act between routing overheads and routing performance, for source routing, the routing overheads are hard to quantify without considering resource reservation. In this paper, we will focus on routing algorithms that do not incur excessive overheads (i.e., single-path routing and multi-path routing that only probes a small number of paths) and ignore the overhead factor. It must be noted, however, that multi-path routing incurs larger overhead than single path routing.

We study the performance of these methods under two link state update policies, the timer based policy and the threshold based policy, compare the effectiveness of the methods in dealing with deterministic imprecision, random imprecision and a combination of both, and identify the advantages and limitations of each algorithm. The main conclusions are the followings. First, at the cost of a higher overhead, dynamic multi-path routing, which probes multiple paths computed based on the inaccurate state information, is effective

in dealing with both random imprecision and deterministic imprecision. Second, randomized routing, which randomly chooses a path from a set of paths computed based on the imprecise global network state information, is ineffective in most cases. Third, static and localized routing offers better performance than the dynamic routing algorithms when the global network state information is extremely imprecise. Fourth, the performance of safety-based routing depends on the characteristics of the imprecision of the global network state information. Safety-based routing is effective in dealing with deterministic imprecision, especially when the state information is precise. Safety-based routing is ineffective in dealing with random imprecision. When the imprecision is mostly random, the performance of the safety-based routing is similar to that of the widest-shortest routing algorithm [7]. Furthermore, when the imprecision is a combination of random imprecision and deterministic imprecision, safety-based routing may result in (much) worse performance than that of the widest-shortest routing algorithm.

The rest of the paper is structured as follows. We describe the related work in Section 2 and present the link state update policies in Section 3. In Section 4, we discuss the QoS routing schemes that tolerate imprecise state information. Section 5 reports the performance study. Section 6 concludes the paper.

II. RELATED WORK

QoS routing has attracted much attention recently. An extensive survey can be found in [5]. A number of QoS routing schemes that deal with the imprecise state information have been proposed [1], [2], [4], [6], [10], [11]. The impact of the imprecise global network state information on the performance of QoS routing algorithms was studied in [1], [11]. Guerin [6] proved a number of important theoretical results on routing in networks with imprecise state information and proposed QoS routing schemes based on probability. In [2], the probability based QoS routing scheme in [6] was materialized to be the safety-based routing that can effectively deal with the deterministic imprecision caused by the link state update policies. The randomized routing scheme to deal with random imprecision was also introduced in [2]. Chen [4] studied multi-path QoS routing, which simultaneously probes multiple paths for each connection request. Nelakuditi [10] proposed localized QoS routing, which computes a set of feasible paths statically and makes routing decisions by selecting a path from the set of feasible paths based on the information maintained locally at each router. In this work, we do not invent new methods to deal with the imprecise global network state information. We compare the effectiveness of safety-based routing, randomized routing, multi-path routing, and localized routing in dealing with deterministic imprecision, random imprecision, and a combination of both, and identify their strengths and weaknesses.

III. MAINTENANCE OF GLOBAL NETWORK STATE INFORMATION

In this paper, we study the timer based update policy and the threshold based update policy [2]. We will assume that the QoS metric is the bandwidth and that the propagation delay is negligible.

- **Timer based link state update policy.** In the timer based policy, each router periodically updates the state of its links to the rest of the network. The *link state update interval* is a parameter of this policy.
- **Threshold based link state update policy.** This policy is characterized by a threshold value (th). Let b^o be the last advertised value of the available bandwidth for a link, b^c be the current available bandwidth, an update is triggered when $\frac{|b^o - b^c|}{b^o} > th$. To control the overheads, a *hold-down timer* is introduced in the threshold based policy. The hold-down timer specifies the minimum time interval between consecutive updates of the same link.

Using the timer based policy, a small link state update interval results in precise state information while a large link state update interval results in imprecise state information. The imprecision resulted from the large update interval is random. The threshold policy allows more accurate link state to be maintained in comparison to the timer based policy since the link state is updated whenever the change of the link state passes the threshold without waiting for the next link state update period. When the hold-down timer is equal to 0, the threshold based policy introduces deterministic imprecision. The imprecision is deterministic in the sense that although the absolute value of the link state cannot be determined, the range of the link state value can be decided. For example, using the threshold based policy with $th = 0.1$, when a link declares that its available bandwidth is $10Mbps$, we know that the bandwidth of that link is in the range of $[9Mbps, 11Mbps]$. A large hold-down timer will introduce random imprecision. By considering these link state update policies, we can evaluate the effectiveness of different methods in dealing with deterministic imprecision, random imprecision and a combination of both.

IV. ROUTING METHODS TO TOLERATE IMPRECISE STATE INFORMATION

This section briefly describes the QoS routing methods to tolerate imprecise state information that we study in the paper. The methods include safety-based routing [2], randomized routing [2], multi-path routing [4], proportional sticky routing (a localized routing scheme) [10], and static multi-path routing. Among these schemes, safety-based routing, randomized routing, and multi-path routing are dynamic schemes that compute feasible paths dynamically based on the current network state information while proportional sticky routing and static multi-path routing do not use the current network state information to make routing decisions. Next, we will describe these methods.

A. Safety-based routing

Safety-based routing algorithms were proposed in [2] to deal with the deterministic imprecision. It does not apply when the timer based policy is used. The idea is to infer the range of the potential available bandwidth and use the range to compute the *safety* of a link, that is, the probability that the link can support the requested bandwidth. Assuming that the hold-down timer is 0, for the threshold based policy with threshold th , let the last advertised bandwidth value be b_o , the range of the potential available bandwidth is in between $(1 - th)b_o$ and $(1 + th)b_o$. Assuming the available bandwidth is uniformly distributed in the range [2], the safety of the link for a connection that requires b_r units of bandwidth is $\frac{(1+th)b_o - b_r}{(1+th)b_o - (1-th)b_o}$, when b_r is in the range $[(1 - th)b_o, (1 + th)b_o]$. When $b_r < (1 - th)b_o$, the link guarantees to support the request, and the safety of the link is 1. When $b_r > (1 + th)b_o$, the link cannot support the request and the safety of the link is 0. Once the safety of each link is determined, the safety of a path is the product of the safeties of all links in the path. Two safety-based algorithms are proposed in [2], *shortest-safest* and *safest-shortest* routing. Safest-shortest routing selects the min-hop path with maximum safety. Shortest-safest routing selects among the safest paths the min-hop path. We will use shortest-safest routing in the evaluation since shortest-safest routing performs better than safest-shortest routing [2].

B. Randomized routing

The idea of randomized routing [2] is to compute a set of feasible paths and then randomly select a path for the connection. Thus, the routing does not always select the “best” path that is computed based on the imprecise global network state information, which offsets the impacts of the imprecision. In this study, we use the per pair path selection heuristic [12] to determine the set of feasible paths. In the experiments, the maximum number of feasible paths is 5.

C. Multi-path routing

Multi-path routing [4] probes multiple feasible paths simultaneously. It also offsets the impact of the imprecision since it not only probes the “best” path. The multi-path routing proposed in [4] is a distributed algorithm. In the study, we use a centralized multi-path routing algorithm. Specifically, we use the per pair path selection heuristic [12] to select up to 5 feasible paths and probe the paths simultaneously. When multiple paths can satisfy the QoS requirement of a connection, the shortest path is selected for the connection.

D. Localized routing

The localized routing algorithm that we consider is the proportional sticky routing (*PSR*) [10]. In the *PSR* scheme, it is assumed that each node has a predefined set of candidate paths to each of the destination nodes. For each connection

request, *PSR* selects a path in the predefined set based on the flow blocking probability. Paths with a lower blocking probability will be selected more frequently. The selection of the predefined set of candidate paths can greatly affect the performance of *PSR*. In our study, we use the global path selection scheme [12] which has been demonstrated to perform well in comparison of other path selection schemes. Once the set of paths is determined, *PSR* basically introduces a heuristic to decide which path should be used for each connection quest.

E. Static multi-path routing

The static multi-path routing always probes the same set of pre-computed paths when a connection request arrives. In the study, we use the same path selection scheme as that used in the *PSR* scheme to compute the set of feasible paths between all pairs of nodes. Essentially, the *PSR* scheme adds intelligence into path selection in order to select the right path for a connection request while static multi-path routing tries out all potential candidates.

V. PERFORMANCE STUDY

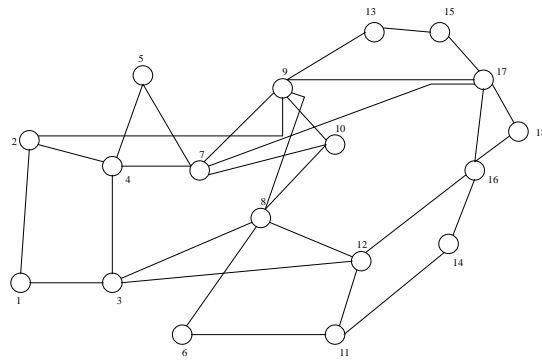


Fig. 1. The ISP topology

This section shows some results of our performance study of the routing algorithms. We carried out extensive simulation on different kinds of “Internet-like” random topologies with up to several hundreds of nodes. The “Internet-like” random topologies are generated using the *Georgia Tech Internetwork Topology Models (GT-ITM)* [13]. The trend in the results for these random topologies are similar to that in the results for an ISP topology shown in Fig. 1. Thus, we will only present the results for the topology shown in Fig. 1. All the links are assumed to be bi-directional and of the same capacity, with C units of bandwidth in each direction. The flow dynamics of the network are modeled as follows. Flows arrive at a node according to a Poisson process with a rate λ . The destination node is chosen randomly from all nodes except the source node. The connection holding time is exponentially distributed with a mean of $1/\mu$ seconds. The offered network load is given by $\rho = \lambda N h' B / \mu L C$, where N is the number of source nodes, L is the number of links, h' is the mean number

of hops per flow, averaged across all source-destination pairs, and B is the average bandwidth requirement for the flows. The parameters used in this simulation are $C = 20$, $N = 18$, $L = 60$. $h' = 2.36$. The mean connection holding time is 60 seconds, that is, $1/\mu = 60$. Unless specified otherwise, the bandwidth requirement of a flow follows an exponential distribution with a mean value of $B = 3$. The average flow arrival rate, λ , is set depending upon the desired load.

Performance of the safety-based routing, multi-path routing, randomized routing and the localized routing is compared with a commonly used dynamic *widest-shortest* routing algorithm [7]. The widest-shortest routing algorithm finds the shortest path that can satisfy the connection requirement. When there are multiple shortest paths available, it selects the one that has the largest available bandwidth for the connection. In the experiments, a blocked flow is dropped without being retried. All the results are obtained with a 95% confidence level.

Timer based link state update

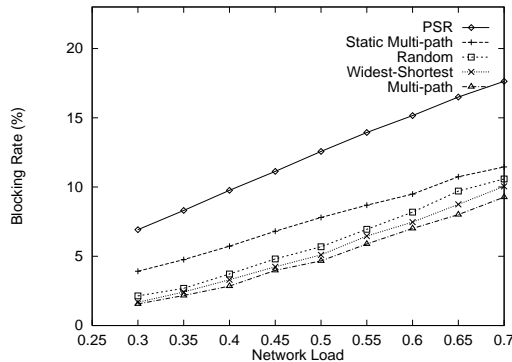


Fig. 2. Timer based policy (update interval = 5)

Fig. 2 shows the results when a timer based link state update policy with a small update interval (5 seconds) is used to maintain the link state. Since the safety-based routing does not apply to this method, we only compare randomized routing, multi-path routing, localized routing, static multi-path routing, and the widest-shortest scheme. Fig. 2 shows the results for the exponentially distributed bandwidth requirement for the flows with a mean value of 3 units. This experiment shows that when the link state update interval is small, the dynamic schemes in general perform better than the schemes, such as localized routing and static routing, that do not use global network state information to make routing decisions. Among the dynamic schemes, multi-path routing performs slightly better than the widest-shortest routing, which in turn, performs slightly better than the randomized method.

Among the two schemes that do not use global network state information, the *PSR* scheme does not work well for flows with exponentially distributed bandwidth requirements. This is because *PSR* infers the current network state using the

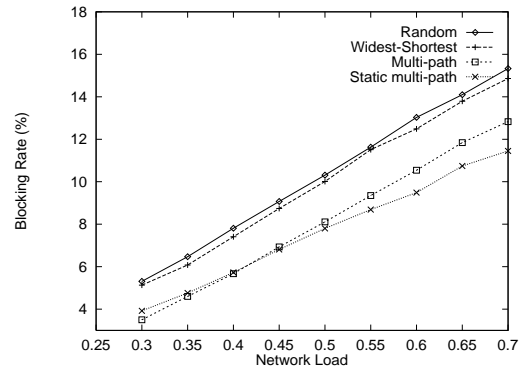


Fig. 3. Timer based policy (update interval = 120)

blocking probability for the given paths without distinguishing the requests with different bandwidth requirements. Thus, when the bandwidth requirement of the flows is exponentially distributed, the network state inferred is very inaccurate and the performance degrades. It can be expected that no *PSR* type of algorithm can perform better than the static multi-path routing algorithm when both algorithms use the same set of initial feasible paths. Actually, in all the experiments we carried out, *PSR* performs worse than static multi-path routing. Thus, in the rest of the paper, we will omit the results for *PSR* and use the static multi-path routing to represent the routing techniques that are not affected by the precision of the global network state information.

Fig. 3 shows the results when a timer based policy with a large update interval (120 seconds) is used. The other parameters are the same as those in Fig. 2. In this case, static multi-path routing performs the best when the network is under heavy load while the dynamic multi-path routing performs the best when the network is under light load. The multi-path schemes are significantly better than the dynamic single path routing schemes, the widest-shortest and the randomized methods, which indicates that multi-path routing is effective in dealing with random imprecision. In this experiment, randomized routing consistently performs worse than the widest-shortest method. Actually, in all the experiments, the randomized method either has the similar performance as the widest-shortest scheme or performs slightly worse than the widest-shortest scheme, which demonstrates that randomly selecting a path from the set of feasible paths computed using the imprecise state information is not effective in dealing with random imprecision. This is because when using the per pair path selection algorithm to compute the set of feasible paths, the lengths of the feasible paths are either the same as the length of the path selected by the widest-shortest routing algorithm or slightly longer. When a path is randomly selected from the set of feasible paths, if the quality of the path is not significantly better than that of the path selected by the widest-shortest routing algorithm, the extra length of the randomly selected path will degrade the overall routing performance.

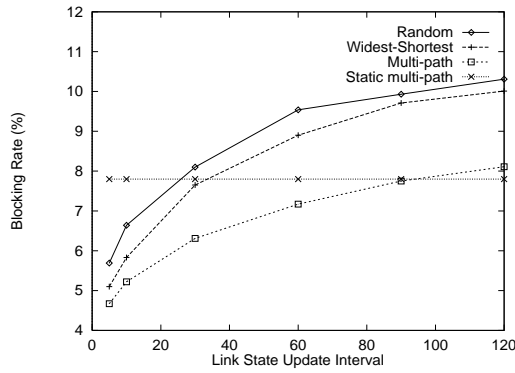


Fig. 4. Impact of link state update interval (load = 0.5)

Fig. 4 shows the impact of the link state update interval. This experiment assumes the network load to be 0.5 and the bandwidth requirement follows an exponential distribution with a mean value of 3 units. As can be seen from the figure, when the link state update interval becomes larger, the performance of all the dynamic routing algorithms degrades. When the timer based link state update policy is used and the network cannot maintain a high link state update frequency, static routing is preferred. Notice that the static algorithm does not use the dynamic link state information to perform routing and thus has the same blocking rate for different link state update intervals.

Threshold based link state update policy

We will first examine the deterministic imprecision resulted from this link state update policy by assuming that the hold-down timer has no effects. Fig. 5 shows the results for the threshold based link state update policy with a small threshold value ($th = 0.1$). Thus, the link state is updated when the available bandwidth changes by 10 percent. The other experimental parameters are the same as those for Fig. 2. As can be seen from the figure, safety-based routing performs noticeably better than the other routing schemes, which shows that selecting routes based of probability is effective when the network state information is precise. The performance of multi-path routing is similar to that of the widest-shortest scheme because the global network state information is precise and the widest-shortest routing algorithm selects routes effectively.

Fig. 6 shows the results for a threshold based policy with a large threshold value ($th = 0.9$). The other experimental parameters are the same as those in Fig. 5. In this case, the link state is updated only when the available bandwidth changes by 90%. As can be seen in the figure, the effectiveness of the safety-based routing decreases in comparison to the case when the threshold is small in Fig. 5. However, safety-based routing performs slightly better than the widest-shortest method and is still the best dynamic uni-path routing method among all the uni-path methods. Multi-path routing performs significantly better than the other algorithms, which

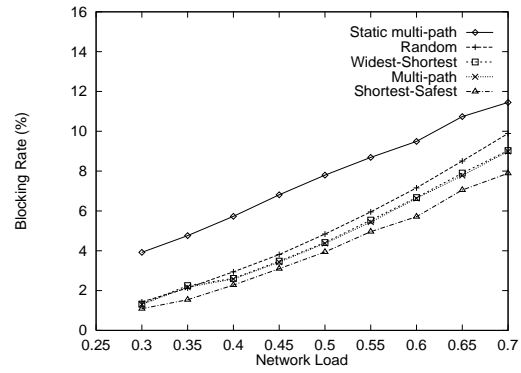


Fig. 5. Threshold based link state update policy ($th = 0.1$, $timer = 0$)

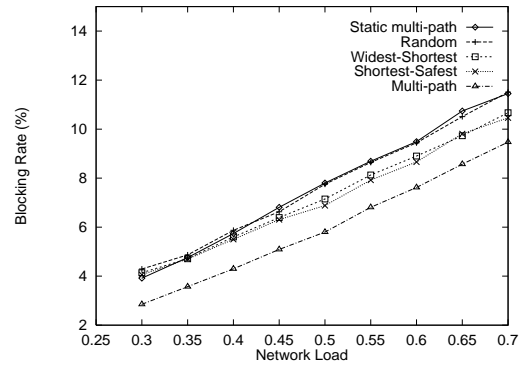


Fig. 6. Threshold based link state update policy ($th = 0.9$, $timer = 0$)

indicates that it is very effective in dealing with deterministic imprecision.

Fig. 7 shows the impact of the threshold on the routing performance assuming that the network load is 0.5. As can be seen in the figure, for all threshold values, safety-based routing consistently performs better than other uni-path routing schemes. This indicates that safety-based routing is effective in dealing with deterministic imprecision. Notice the crossing of curves for the multi-path method and the safety-based method. It shows that the multi-path method can tolerate high deterministic imprecision better than the safety-based

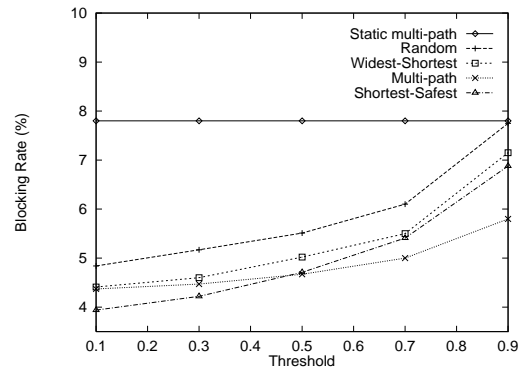


Fig. 7. Impact of the threshold, (Load = 0.5, timer = 0)

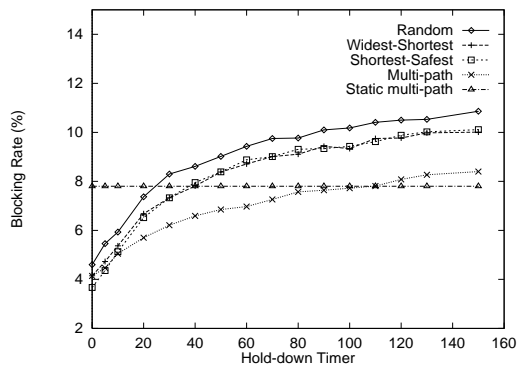


Fig. 8. Impact of the hold-down timer ($th = 0.1$, $load = 0.5$)

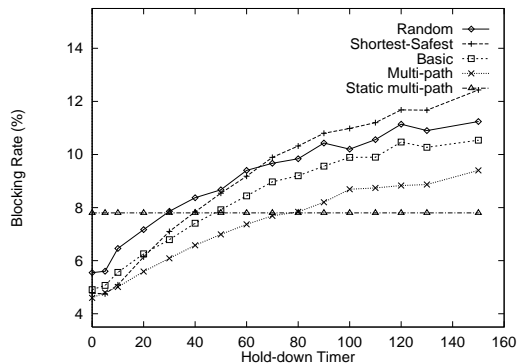


Fig. 9. Impact of the hold-down timer ($th = 0.5$, $load = 0.5$)

method.

Fig. 8 shows the impact of the hold-down timer when the threshold is small ($th = 0.1$). The experiment assumes $load = 0.5$. As the hold-down timer becomes larger, the global network state information becomes increasingly randomly imprecise. As can be seen in the figure, safety-based routing is effective only when the hold-down timer is small. When the hold-down timer is larger than 20 seconds, safety-based routing has the similar performance as the widest-shortest algorithm. Dynamic multi-path routing is more effective than other dynamic methods. When the hold-down timer is larger than 110 seconds, static multi-path routing yields the best performance.

Fig. 9 shows the impact of the hold-down timer when the threshold is larger ($th = 0.5$). This experiment assumes $load = 0.5$. In this case, the global network state information is very imprecise. Both random imprecision and deterministic imprecision are involved. Under this condition, the performance of safety-based routing degrades much faster than all other routing schemes as the hold-down timer becomes larger. When the hold-down timer is large, safety-based routing yields the highest blocking rate. Computing paths based on imprecise safety yields worse results than computing paths based on imprecise link state value. Hence, safety-based routing cannot tolerate the combination of large random imprecision

and large deterministic imprecision. This experiment also shows that when the global network state information is imprecise, static multi-path routing becomes more appealing.

VI. CONCLUSIONS

In this work, we investigated the QoS routing schemes that tolerate imprecise link state information. Five routing methods, namely safety-based routing, randomized routing, multi-path routing, localized routing, and static multi-path routing, are considered. The interaction between the routing algorithms with the timer based policy and the threshold based policy is studied. The conclusions are the followings. First, at the cost of higher routing overheads, multi-path routing is effective in dealing with both random imprecision and deterministic imprecision. Second, randomized routing is ineffective in most cases. Third, static and localized routing offers better performance than the dynamic routing algorithms when the global network state information is extremely imprecise. Fourth, the performance of safety-based routing depends on the characteristics of the imprecision of the global network state information. Safety-based routing is effective in dealing with deterministic imprecision and ineffective in handling random imprecision. Furthermore, safety-based routing may result in poor routing performance when the imprecision is a combination of random imprecision and deterministic imprecision. These conclusions suggest that to design practical QoS routing algorithms, the characteristics of the global network state information maintained at each router must be studied.

REFERENCES

- [1] G. Apostolopoulos, R. Guerin, S. Kamat and S. Tripathi "Quality of Service Based Routing: A Performance Perspective", *ACM SIGCOMM*, 1998.
- [2] G. Apostolopoulos, R. Guerin, S. Kamat, and S. Tripathi, "Improving QoS Routing Performance Under Inaccurate Link State Information." *Proceedings of the 16th International Teletraffic Congress*, June 7-11, 1999.
- [3] Baruch Awerbuch, Yi Du, Bilal Khan, and Yuval Shavitt, "Routing Through Networks with Hierarchical Topology Aggregation", *Journal of High Speed Networks*, 7(1):57-73, 1998.
- [4] S. Chen and K. Nahrstedt, "Distributed QoS Routing with Imprecise State Unformation." *ICCCN'98*, October 1998.
- [5] S. Chen and K. Nahrstedt, "An Overview of Quality-of-Service Routing for the Next Generation High-Speed Networks: Problems and Solutions," *IEEE Networks*, Special Issue on Transmission and Distribution of Digital Video, Nov./Dec., 1998.
- [6] R. Guerin and A. Orda, "QoS-based Routing in Networks with Inaccurate Information: Theory and Algorithms.", *IEEE INFOCOM'97*, April 1997.
- [7] Q. Ma and P. Steenkiste, "Quality-of-Service Routing with Performance Guarantees", *4th IwQoS'97*, May 1997.
- [8] G. Malkin, "RIP Version 2.", RFC 2453, November 1998.
- [9] J. Moy, "OSPF Version 2", RFC 2328, April 1998.
- [10] Srihari Nelakuditi, Zhi-Li Zhang, and Rose P. Tsang, "Adaptive Proportional Routing: A Localized QoS Routing Approach", *In IEEE Infocom*, April 2000.
- [11] A. Shaikh, J. Rexford and K. Shin, "Evaluating the Overheads of Source-Directed Quality-of-Service Routing", *International Conference on Network Protocols (ICNP)*, 1998.
- [12] X. Yuan and A. Saifee, "Path Selection Methods for Localized Quality of Service Routing." *IC3N'01*, Oct. 2001.
- [13] Ellen W. Zegura, Ken Calvert and S. Bhattacharjee. "How to Model an Internetwork." *Proceedings of IEEE Infocom '96*, 1996.