A Stability-Based Link State Updating Mechanism for QoS Routing*

Miao Zhao[1], Huiling Zhu[1], Victor O. K. Li[1], Zhengxin Ma[2]

[1]Department of Electrical and Electronic Engineering
The University of Hong Kong, Hong Kong
{mzhao, hlzhu, vli@eee.hku.hk}

[2]Department of Electronic Engineering
Tsinghua University, Beijing, China
{mazhengxin@sat.mdc.tsinghua.edu.cn}

***Abstract*— QoS routing which satisfies diverse application requirements and optimizes network resource utilization needs accurate link states to compute paths. Suitable link state update (LSU) algorithms which ensure timely propagation of link state information are thus critical. Since traffic fluctuation is one of the key reasons for link state uncertainty and the existing approaches can not effectively describe its statistical characteristics, in this paper, we propose a novel stability-based (SB) LSU mechanism which consists of a second-moment-based triggering policy and a corresponding stability-based routing algorithm. They incorporate knowledge of link state stability in computing a stability measure for link metrics. With extensive simulations, we investigate the performance of SB LSU mechanism and evaluate its effectiveness compared with existing approaches. Simulation results show that SB LSU can achieve good performance in terms of traffic rejection ratio, successful transmission ratio, efficient throughput and link state stability while maintaining a moderate volume of update traffic.**

***Index Terms*— QoS routing, link state update, stability.**

I. Introduction

QOS routing is one of the key mechanisms for QoS guarantees. It consists of two basic tasks[1]. The first task is to collect the state information and keep it up-to-date. The second one is to find a feasible path for a new connection based on the collected information. The performance of a routing algorithm depends on how well the first task is performed. Due to inherent network uncertainties such as the delay of the physical transmission media, dynamics of network resources, finite update frequency, aggregation in large networks, hidden information of private networks and approximate calculations of the system parameters and values [2], the state information of network resources collected by a router is generally inaccurate to some extent. This inaccuracy will adversely affect the efficiency and reliability of routing.

How do we solve this problem? One effective way is to update link state information whenever there is a change of link state such as the available bandwidth, but this requires excessive number of update messages. But if we do not perform updates whenever there are link state changes, how often should we do it? Most existing proposals belong to one of four alternative methods. They are period-based updates, threshold-based updates, equal class based updates and unequal class based updates[1]. We will further discuss these four policies in Section II. Previous researchers have concluded that the above triggering policies may be effective in responding to significant changes in available bandwidth to some extent, but there are some drawbacks as well [1][3]. Propagating link state update messages in a periodic fashion can be considered a static triggering policy. It can not ensure timely propagation of significant changes of link state metrics. Although the other three approaches are dynamically triggered, they are based on boundary crossing and can not reflect the dynamic characteristics of bandwidth fluctuations within a class. Actually this description of fluctuation is essential to QoS network because traffic fluctuation is one of the key reasons for link state uncertainty. Without accurate description of traffic flunctuation, network performance will be severely degraded, especially for those QoS mechanisms with the prerequisite hypothesis that the link state is always stable. Since traffic flunctuation is unavoidable and due to the inherent limitation of existing approaches based on boundary crossing, we propose the stability-based link state update (SB LSU) scheme in this paper.

In order to minimize significant fluctuations and effectively maintain link state stability, we propose the stability-based link state update (SB LSU) scheme which can clearly reflect the characteristics of link state fluctuations. In this scheme, we concentrate on keeping the link state stable by informing all the nodes in the network when the stability of a link changes. The probability density function (PDF) is used to describe the stochastic characteristics of a random variable representing the link state. However, it is difficult and complex to acquire the exact PDF, so we propose a second-moment-based update scheme. In addition, we note that existing routing algorithms are always isolated from the triggering policy. But they are actually dependent on and interact with each other and should be considered collectively. In this paper, we propose a novel stability-based routing algorithm based on the triggering policy

* This research is supported in part by the Areas of Excellence Scheme established under the University Grants Committee of the Hong Kong Special Administrative Region, China (Project No. AoE/E-01/99).

0-7803-8938-7/05/$20.00 (C) 2005 IEEE

of our SB LSU mechanism.

The rest of this paper is organized as follows. In Section II, we present four LSU algorithms and discuss their performance. Then in Section III, we propose SB LSU which uses the second-moment parameter as the measurement metric in the LSU triggering policy. In Section IV, we present our simulation results. Conclusions and suggestions for future work are presented in Section V.

II. Previous Work

One of the primary requirements in QoS routing is updating the network state information so it is available for path computation and selection. There are various methods to trigger the updates. Due to space limitations, in this paper, we concentrate on the following four alternatives that cover a wide range of different options and operational characteristics. All the following schemes select available bandwidth of a link or a node interface as their link state metric.

- Period-based updates (PB): it is a very simple mechanism but this approach can not ensure timely propagation of significant changes, and therefore can not guarantee accurate information for path computation[3]. When the network is highly dynamic and changes are frequent, in order to learn about the network changes in a timely manner,smaller time interval of triggering period is required which will bring much additional overhead and may also be impractical.
- Threshold-based updates (TB): in this algorithm, there is a constant threshold value (th) to distinguish significant changes from minor ones. For an interface i of a node, bw_i^o is the last advertised value of the link state, and bw_i^c is the current link state value. An update is triggered when $|bw_i^o - bw_i^c| / bw_i^c > th$ for $bw_i^o > 0$. For $bw_i^o = 0$, an update is always triggered. This algorithm effectively solves the problem we mentioned in period-based updates since it tends to provide more detailed information when operating with smaller available bandwidth and becomes progressively less accurate for larger values of available bandwidth.
- Equal class based updates (ECB): this algorithm is characterized by a constant B that is used to partition the available bandwidth operating region of a link into multiple equal sized classes: $(0, B)$, $(B, 2B)$, $(2B, 3B)$, $\ldots$, etc[1]. An update is triggered when a change in available bandwidth moves the operating region to a different class from the one it belongs to at the previous time slot. In this mechanism, the same weight is assigned for all ranges of available bandwidth.
- Unequal class based updates (UCB): this policy is characterized by the combination of two parameters B and f which are used to define unequal size classes: $(0, B)$, $(B, (f+1)B)$, $\left((f+1)B, \left(f^2+f+1\right)B\right)$, $\ldots$, etc. Unlike the equal class based algorithm, the class size grows geometrically by the factor f. An update is triggered as before when a class boundary is crossed. This policy has fewer and larger classes when the available bandwidth is high and smaller classes when the available bandwidth is low. Consequently, it provides a more detailed and accurate state description for the low bandwidth region [3]. This is reasonable because we need more exact information when the available resource is scarce.

Class based policies are based on boundary crossing and can not reflect the bandwidth fluctuation within one class. In these policies, triggering an update each time the available bandwidth crosses a class boundary may cause oscillations when the available bandwidth value fluctuates around a class boundary [1]. Under this scenario, the performance of the class based updates will be drastically degraded.

Much work has been done to compare the network performance of the above four methods under different network topologies by simulation [3]. None of them considers link state stability which is very important in QoS performance. Considering that the fluctuation of traffic flows is one of the main reasons causing the uncertainty of link state, we intend to explore a scheme which is based on the characteristics of link state fluctuations. SB LSU proposed in this paper is based on statistical characteristics and the trigger policy is according to the change of stability characteristics of selected state metrics. In the following sections, we will introduce our proposed mechanism and compare it with the four existing LSU algorithms.

III. Methodology

As we have mentioned above, in a statistical guaranteed QoS mechanism, link state stability is of great importance. The absence of stability control and the random admission of new connections will cause traffic fluctuations in the network. Many researchers have discussed the impact of such fluctuations on network congestion [4]. Thus in order to enhance the performance of the whole network, we focus on the stability maintenance of the link state. In this section, we will present our proposed SB LSU algorithm which is based on the stability characteristics of the link state. For ease of comparison with previous schemes, we choose the available bandwidth as the link state metric. Our scheme mainly comprises two parts: one is a second-moment-based triggering policy, and the other is a stability-based routing algorithm.

A. Triggering Policy

We propose a state update scheme based on changes in state stability, and the state variance is chosen to reflect the fluctuation of link stability. It is reasonable to assume that when more resource is available, more fluctuations could be tolerated. Thus, when the available bandwidth is selected as the link state metric, we choose the following metric to describe the change of link state:

$$F\left(\mu, \sigma^2\right) = \frac{\sigma^2}{(C-\mu)} / C \tag{1}$$

where σ^2 and μ are the variance and the expectation of the traffic rate on a link, C is the link capacity, and $C - \mu$ is the mean of the available bandwidth of the link.

Our scheme is a second-moment-based mechanism because its triggering policy is characterized by the function $F\left(\mu, \sigma^2\right)$. Hence we call this function a second-moment parameter. When the value of the second-moment-based function exceeds a certain threshold, updates are triggered. With a certain threshold, when the available bandwidth is abundant, more fluctuation of variance can be tolerated. On the contrary, with low available bandwidth, the sensitivity of our triggering policy will be significantly enhanced. As a consequence, it also tends to provide a more refined description for the low bandwidth region. Therefore, our scheme is capable of self adaptation.

In addition, in existing class based policies, triggering an update each time the available bandwidth value crosses a class boundary has the undesirable effect of generating meaningless update messages when the available bandwidth fluctuates around a class boundary [1]. SB LSU algorithm can effectively dampen such oscillatory behavior by adopting the second-moment parameter rather than the absolute value of the available bandwidth.

B. Stability-based Routing Algorithm

Some previous work propose heuristic QoS routing algorithms, such as widest-shortest path, shortest-widest path, short-distance path, and dynamic-alternative path [5]. Performance analysis indicates that from among the many heuristics proposed for routing requests with bandwidth requirements, the shortest (with respect to the hops in the path) path heuristics perform better than the widest path heuristics[1]. Thus we will select widest-shortest path heuristics as the basis of our proposed stability-based routing algorithm. The mechanism of widest-shortest path is as follows: Links that have insufficient available bandwidth for the request that is being routed are pruned from the network before the path is computed. Then the minimum hop counts between the source and destination are discovered and the widest one is chosen.

In our second-moment-based triggering policy, the latest mean rate μ and variance σ^2 of occupied bandwidth can be acquired via update messages. Since our objective is to keep the link state stable, we choose the coefficient of variation σ^2/μ^2 as the stability parameter. We use the average stability of all links along the path to represent the path stability. The path stability can then be handled as other path metrics. In correspondence to our triggering policy, we can define a stability-based routing algorithm as follows:

- First compute all the paths which satisfy the QoS requirements;
- Find the paths with the minimum number of hops. If only one shortest hop path exists, following the dynamic-alternative path algorithm, the paths with one hop more than that of the minimum hop path and which meet the required QoS will also be considered;
- The path with the maximum stability is selected.

Such QoS routing considering multiple paths has many merits. [6] indicates that multiple routing offers a significant improvement over the single-path approach, especially if the link state information is outdated and inaccurate. In addition, hop-based algorithms win over bandwidth-based algorithm.

IV. Results

A. Simulation Environment

We aim to evaluate the QoS performance under a given network environment. By collecting statistics on some parameters, we can observe how the different LSU algorithms perform with different parameter values. The topology used in our simulation experiments is shown in Figure 1. This MCI topology is popular in many QoS routing studies. In Figure 1, R_0, R_1, ..., R_{17} are core routers. The neighboring routers are connected via physical links. The total number of links N_3 is 60. S_0, S_1, ..., S_{17} are subnets which play the role of traffic sources. It is assumed that the delay between a subnet and its connected router can be ignored, and all links between routers are symmetric and have the same bandwidth of 1Mb/s. The queue size of each interface in each router is 200kbits.

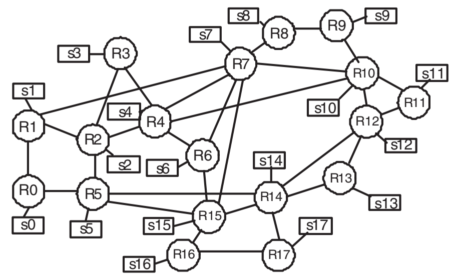

Fig. 1. MCI Topology used in the simulation experiments

The symbols used in our simulation are shown in Table I.

TABLE I

SYMBOL LIST

Symbol	Explanation
C	the link capacity of each link in the network
N_1	total number of subnets in the simulated MCI topology
N_2	the average total number of traffic connections from one subnet during the simulation
N_3	the total number of links in the simulated MCI topology
N_p	the number of nodes along the path p
p_i	the ith link on path p
r	the average traffic rate of the aggregated traffic from one subnet to its attached router
h	the total number of hops of one selected path
N_{rej}	the total number of rejected traffic connections
N_{fail}	the total number of unsuccessful traffic connections

In our simulation, each subnet S_i generates connection requests according to a Poisson process with rate λ_1, and with uniform random selection of destination nodes. The holding time for each connection is a constant τ. We model each traffic connection as an ON-OFF traffic source with mean rate of λ_2. The average traffic rate of the aggregated traffic from one subnet to its attached router is denoted as r.

$$r = \lambda_1 \cdot \tau \cdot \lambda_2 \tag{2}$$

We use the classical Dijkstra's algorithm to find the paths with minimum hops and the mean distance (in number of hops) between nodes is $\overline{h}$. Thus, the network load is calculated as Eq. (3):

$$\rho = \frac{r \cdot N_1 \cdot \overline{h}}{N_3 \cdot C} \tag{3}$$

B. Performance Measures

The performance measures of interest in this paper are routing performance and the volume of update traffic for different LSU algorithms. Routing performance is measured by traffic rejection ratio, failed transmission ratio, and throughput of the network. We measure link state update traffic by the number of update messages generated in the network. This volume includes all the link state update messages generated by all routers over the duration of the simulation. Another measure we will focus on is the stability of specific links. The definitions of these properties respectively goes as follows.

- Traffic rejection ratio

$$P_{rej} = \frac{num\ of\ rejected\ connections}{num\ of\ offered\ connections} = \frac{N_{rej}}{N_1 \cdot N_2} \quad (4)$$

- Failed Transmission ratio

$$\begin{aligned} P_{tran\_fail} &= \frac{num\ of\ failed\ connections}{num\ of\ admitted\ connections} \\ &= \frac{N_{fail}}{N_1 \cdot N_2 \cdot (1 - P_{rej})} \end{aligned} \quad (5)$$

In our simulation, each traffic connection has a specified maximum packet loss probability 10^{-3}. If the packet loss probability is greater than this value during the transmission, we consider the transmission of this traffic connection has failed. Otherwise, the transmission is successful.

- The throughput of the network refers to the successfully transmitted traffic connections.

$$G_{success} = N_1 \cdot N_2 \cdot (1 - P_{rej}) \cdot \left(1 - P_{tran\_fail}\right) \quad (6)$$

  where $(1 - P_{rej}) \cdot \left(1 - P_{tran_fail}\right)$is called the effective throughput ratio.
- Update traffic: the total volume of state update messages generated in the whole network;
- Link burstiness: the burstiness of the aggregated traffic flow on one link which is in terms of the coefficient of variation σ^2/μ^2;
- Path burstiness: the average burstiness of all links along the path $\frac{1}{N_p} \sum_{i=1}^{N_p} \left(\sigma_{p_i}^2 / \mu_{p_i}^2\right)$.

C. Simulation Results and Analysis

In this subsection, we review how the choice of parameters of the triggering policy affects both the QoS routing performance and the update traffic volume.

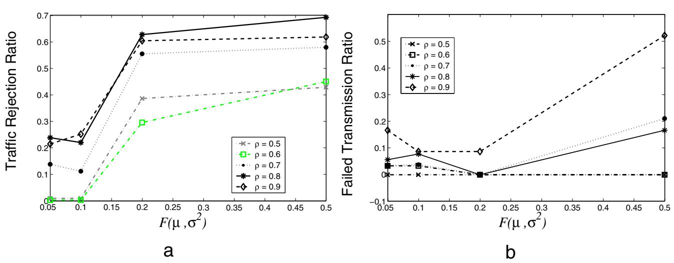

Fig. 2. a.Traffic rejection ratio vs. SB values b.Failed transmission ratio vs. SB values

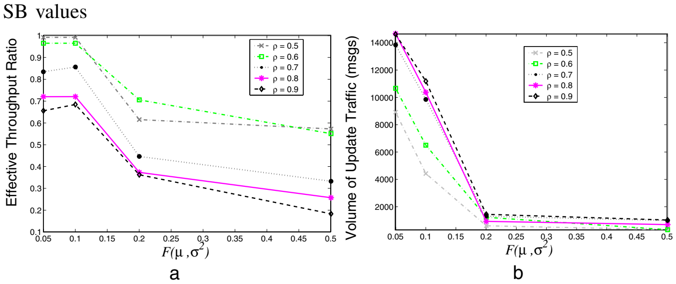

Fig. 3. a.Efficient throughput ratio vs. SB values b.Update traffic comparison vs. SB values

Fig. 2 shows the traffic rejection ratio and the failed transmission ratio under different SB parameters. Fig. 3.a shows the effective throughput ratio with different SB values, and the volume of update traffics versus the SB values are shown in Fig. 3.b. The results recorded in Fig. 2 - 3 show that the threshold value for the second-moment parameter has great impact on the QoS performance. Optimal throughput can be achieved with a threshold no more than 0.1, which is close to the ratio of the traffic mean rate to the link capacity. We should take it into consideration in future simulations to make certain what and to what extent it impacts routing performance.

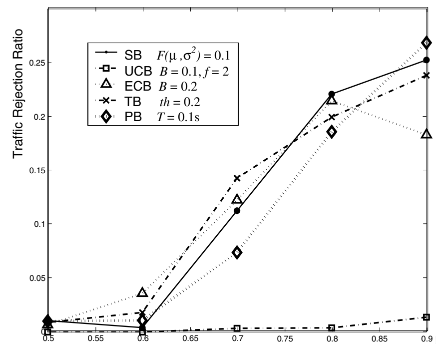

Fig. 4. Traffic rejection ratio comparison of various LSU algorithms

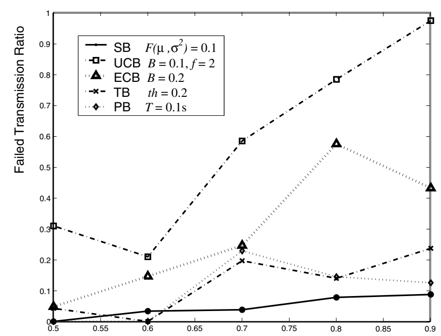

Fig. 5. Failed transmission comparison of various LSU algorithms

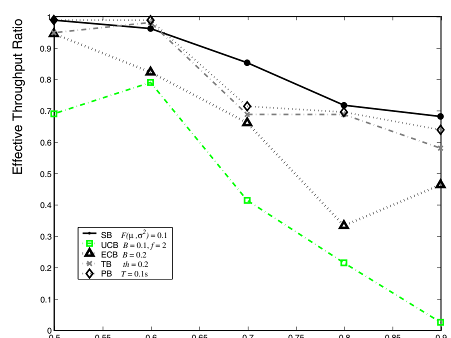

Fig. 6. Efficient throughput ratio comparison of various LSU algorithm

Among the many criteria of evaluating network performance, throughput is one of the most important ones. So achieving higher throughput is our ultimate goal. The parameter setting for each LSU mechanism except SB in our simulation is according to the results presented in [3]. For each algorithm, we select the setting which achieves the best performance and cost combination. The results recorded in Fig. 4, 5 and 6 show that the traffic rejection ratio of SB LSU

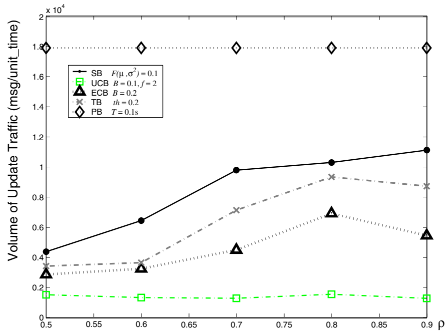

Fig. 7. Update traffic comparison of various LSU algorithms under different traffic loads

can be dynamically adjusted according to the update messages so that QoS routing may select a comparatively optimal path to circumvent link congestion and severe traffic fluctuation. As a consequence, its throughput outperforms other approaches. In addition, with the optimal choice of parameters, TB and PB will probably give the best performance. Since UCB involves two factors and f can greatly widen the range of a class, the cost (traffic updates) of UCB and its traffic rejection ratio are much lower than other approaches. However, this also implies that the failed transmission ratio will increase dramatically.

[7] indicates that long periods and coarse triggers result in stale link state information which will cause suboptimal path selection, or even failure to select a route which accommodates a new connection. Hence, tuning the frequency and the volume of link state updates require a thorough understanding of the trade-offs between network overhead and the accuracy of routing information. Fig. 7 shows the volume of update messages of various LSU algorithms under different traffic loads. It is found that SB LSU achieves higher throughput at the expense of slightly larger volume of update traffic when compared with UCB, ECB and TB. For SB, TB and class based triggering, update traffic is significantly reduced as the triggering policies become less sensitive.

TABLE II

BURSTINESS COMPARISON OF TWO PATHS BETWEEN R_{11} AND R_0 WITH VARIOUS LSU ALGORITHMS

		Coefficient of variation				
ρ	LSU	R_{11}, R_{10}	R_{10}, R_7	R_7, R_1	R_1, R_0	Ave
0.6	SB	0.0075	0.0123	0.0127	0.0196	0.013
0.6	UCB	0.1779	0.0106	0.0338	0.0914	0.078
0.6	ECB	0.0470	0.0271	0.0666	0.0423	0.046
0.6	TB	0.0424	0.0172	0.0520	0.0849	0.049
0.6	PB	0.0325	0.0164	0.0316	0.0724	0.038
ρ	LSU	R_{11}, R_{12}	R_{12}, R_{14}	R_{14}, R_5	R_5, R_0	Ave
0.6	SB	0.0444	0.0302	0.0219	0.0211	0.029
0.6	UCB	0.2523	0.0478	0.0053	0.0231	0.082
0.6	ECB	0.0796	0.0241	0.0603	0.0276	0.048
0.6	TB	0.0440	0.0249	0.0293	0.0655	0.041
0.6	PB	0.0334	0.0266	0.0158	0.0840	0.040

Table II shows the comparison of the stability of two selected paths between R_{11} and R_0 with different LSU algorithms in terms of the burstiness of the aggregated traffic flow travelling along the paths. We investigate burstiness on each individual link, identified by its end nodes $[R_i, R_j]$ and take the average burstiness of all the links along the path as the path stability parameter. We find that for both paths, the stability of traffic connections with SB LSU is much higher than with other approaches. SB LSU effectively maintains the link stability which is critical for optimizing routing computation and balancing traffic loads throughout the entire network.

V. CONCLUSIONS

In this paper, we propose a novel stability-based link state update mechanism which mainly consists of two parts: second-moment-based triggering policy and stability-based routing algorithm. The overall goal is to improve network performance by the maintenance of link state stability because the latter is the commonly accepted premise of statistical QoS guarantees. We have already presented a preliminary simulation-based study of the effects of SB LSU algorithm on the performance and cost of QoS routing. In addition, we also investigate four existing LSU algorithms and compare them with our proposal.

With extensive simulations, we verify that the performance of SB LSU is superior to that of other approaches:

- In SB, the dynamic adjustment between access and successful transmission ratio leads to improved throughput;
- In SB, satisfactory routing performance can be achieved while maintaining moderate volume of update messages;
- In SB, the burstiness of the aggregated traffic flow on one link is lower than that of other approaches in terms of the coefficient of variation of the available bandwidth.

In the future, we attempt to improve our scheme, particularly in parameter selection. We will also verify the significance of link state stability in QoS routing from a theoretical point of view. We also plan to have a more systematic comparative analysis of various algorithms under diverse network topologies and with different traffic source models. In addition, new LSU algorithms and stability control mechanisms will be investigated.

REFERENCES

[1] 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 (ITC'16)*, Edinburgh, United Kingdom, Jun 7-11, 1999.

[2] D. Lorenz, and A. Orda, "QoS routing in networks with uncertain parameters," *IEEE/ACM Trans. on Networking,* Vol. 6, No. 6, December 1998, pp. 768-778.

[3] Z. Ma, P. Zhang, and R. Kantola, "Influence of link state updating on the performance and cost of QoS routing in an intranet," *Proceedings of the IEEE Workshop on High Performance Switching and Routing 2001,* Dallas, Texas, USA, May 29-31, 2001, pp. 375-379.

[4] T. Gyires, "Methodology for modeling the impact of traffic burstiness on high-speed networks," *Proceedings of the IEEE International Conference on SMC'99,* Tokyo, Japan, Vol. 1, October 12-15, 1999, pp 980-985.

[5] Q. Ma, and P. Steenkiste, "Quality-of-Service routing for traffic with performance guarantees," *Proceedings of IFIP 5th international workshop on Quality of Service*, New York, USA, 1997, pp. 115-126.

[6] Y. Jia, I. Nikolaidis, and P. Gburzynski, "Multiple path routing in networks with inaccurate link state information," *Proceedings of IEEE International Conference on Communications (ICC'2001)*, Helsinki, Finland, Vol. 8, Jun 11-15, 2001, pp. 2583-2587.

[7] A. Shaikh, J. Rexford, and K. Shin, "Evaluating the overhead of source-directed Quality of Service routing," *Proceedings of the 6th International Conference on Network Protocols (ICNP'1998)*, Austin, Texas, USA, October 13-16, 1998, pp 42-51.