Routing-Oriented Update SchEme (ROSE) for Link State Updating

Nirwan Ansari, Gang Cheng, and Nan Wang

***Abstract*—Few works have been reported to address the issue of updating link state information in order to effectively facilitate Quality-of-Service (QoS) routing. The idea of modeling the QoS link state information as random variables has been reported, but none of the existing works have provided a comprehensive probabilistic approach to link state update that takes the probability density functions of both the user's QoS requirements and the network's QoS measurements into account. We propose the Routing-Oriented update SchEme (ROSE) that utilizes the knowledge of the history of network operations and user's QoS requirements to improve the efficiency of link state update without increasing the network overhead. ROSE is a new class-based link state update scheme which intelligently determines class sizes to minimize the impact of inaccurate link state information. Through theoretical analysis and extensive simulations, we demonstrate that ROSE outperforms other class-based link state update policies.**

***Index Terms*—Quality of Service (QoS), routing, link state update.**

I. Introduction

The ability to provide Qualify of Service (QoS) is a necessity for the next generation integrated networks. Today, QoS routing has become the fundamental focus of study. The goal of QoS routing is to find a path that satisfies multiple QoS constraints while maximizing the network utilization and minimizing users' costs. QoS routing in general consists of two critical issues: link state dissemination and route selection [1]. The link state dissemination addresses how the link state information is exchanged throughout the network; while the route selection elaborates on how to find the optimal path given the available link state information. Many works have addressed the issue of route selection [2]-[7]. In this paper, we concentrate on the issue of link state dissemination. The purpose of link state dissemination is to provide the knowledge of QoS status of all the links to the routing devices (e.g., routers) in a network. Based on this knowledge, the network can then determine the best route for any given end-to-end connection to meet its QoS requirements and utilize the overall network resource efficiently. In order to provide the knowledge of all the QoS parameters of each link, each link itself must employ some scheme to report its own QoS parameters, referred to as "link state update". Generally, it is impractical to assume that routing devices have accurate link state information of all links at all time, because this would require rapid link state updates from all links, hence consuming a large amount of network resource. Therefore, an effective link state update algorithm is necessary to provision QoS. Link state update determines the behavior of how each node updates its status to the entire network, including when to update and how to update. A widely used link state update protocol, OSPF [8], which has also been adopted in many types of networks such as optical networks [9], recommends the link state to be updated once every 30 minutes. However, because of the highly dynamic nature of the traffic, updating in such a long time interval will result in stale/outdated link state parameters. This will compromise the efficiency of QoS routing. Several other link state update policies, such as threshold, equal class and exponential class based update policies [10], have been proposed. In the threshold policy, an update is triggered when the difference between the current value and the previously updated value of a certain parameter exceeds a threshold. That is, given a threshold value τ, an update occurs when $|b_c - b_0| > \tau$, where b_0 is the previously updated value and b_c is the current value of a QoS parameter.

In the equal-class and the exponential-class based update policies, the values of QoS parameters are divided into classes. An update is triggered when the current value of a QoS parameter changes from one class to another. For example, in a two-class situation, if the range (interval) of the first class is $(0, b_1)$, and the range of the second class is (b_1, b_2), then an update will happen when b_c changes from $0 < b_c < b_1$ to $b_1 < b_c < b_2$, or vice versa. What separates the equal class based link state update policy from the exponential class based link state policy is the choice of the boundaries, or in other words, the partitioning of each class. In the equal class based link state update, the class of a QoS parameter is partitioned into equal-sized intervals, for example, *(0,B), (B, 2B), (2B, 3B)*, $\ldots$, etc.. In the exponential class based update, the classes are partitioned into unequal-sized ranges, *(0,B), (B, (f+1)B), ((f+1)B,* $(f^2+f+1)B)$, $\ldots$, etc., whose sizes grow geometrically by a factor of f, where B is a predefined constant. No matter which link state update policy a network adopts, it is unavoidable that the QoS parameters of each node known to the entire network might not be exactly accurate at any given time, due to the staleness and coarse classes. As a result, false routing is inevitable. Some works have been done in analyzing the effect of stale or inaccurate link state information, and attempting to reduce its impact. In [11], extensive simulations were made to uncover the effects of the stale link state information and random fluctuations in the traffic load on the routing and setup overheads. In [12]-[13],

Paper approved by T.-S. P. Yum, the Editor for Packet Access and Switching of the IEEE Communications Society. Manuscript received September 27, 2006; revised June 11, 2007. This work has been supported in part by the National Science Foundation under grant 0435250.

The authors are with the Advanced Networking Laboratory, ECE Dept., NJIT, Newark, NJ 07012, U.S.A. (e-mail: ansari@njit.edu).

Digital Object Identifier 10.1109/TCOMM.2008.060548.

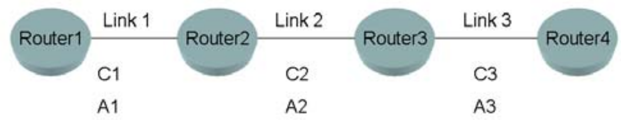

Fig. 1. Illustration of concave and additive constraints: Concave QoS parameters of link 1, 2, and 3 = $C1, C2$, and $C3$. Additive QoS parameters of link 1, 2, and 3 = $A1, A2$, and $A3$. The path is acceptable if $\min\{C1, C2, C3\} \geq C0$ and $A1 + A2 + A3 \leq A0$, where C0 and A0 are required concave and additive constraints.

the effects of the stale link state information on QoS routing algorithms were demonstrated through simulations by varying the link state update interval. A combination of the periodic and triggered link state update is considered in [14]. Instead of using the link capacities or instantaneous available bandwidth values, Li *et al.* [15] used a stochastic metric, Available Bandwidth Index (ABI), and extended BGP to perform the bandwidth advertising.

In this paper, for the purpose of saving network resources and reducing the staleness of link state information, we introduce a new link state information update scheme, Routing-Oriented update SchEme (ROSE)[1]. The uniqueness of ROSE is that it takes the QoS requirements of applications and the network QoS behavior into account. As reviewed above, most of the existing link state update schemes do not consider both the statistical distributions of the actual user's QoS requirements and the network's QoS behavior. In fact, we have discovered that the knowledge of the distribution of the user's QoS requirements and the history of network's QoS behavior can greatly improve the efficiency of link state update and the accuracy of QoS routing. The statistical distribution of the user's QoS requirement and the network's QoS behavior can be obtained from the network operation history. The key concept of ROSE is to utilize these statistical distributions and design a class-based link state update scheme that is able to provide the most helpful link state information for the connection setup processes, hence yielding better performance than other existing link state update schemes. Via theoretical analysis and simulations, we show that ROSE greatly outperforms the state of the art. The rest of the paper is organized as follows. Section II describes the properties of various types of QoS constraints. Section III defines the term "false routing" and the cost of false routing. Then, in Section IV, we describe our proposed efficient link state information update scheme, ROSE. The simulation results are presented in Section V. Finally, concluding remarks are given in Section VI.

II. Properties of QoS Constraints

Most of the QoS constraints (e.g., bandwidth, delay) can be categorized into the following three types: concave, additive, and multiplicative. Multiplicative constraints can be converted into additive constraints by using the logarithm operator. Therefore, only concave and additive constraints are considered in the study of QoS routing. A concave constraint works as follows: in the case of a multi-link end-to-end path, as long as the smallest (or largest) QoS parameter among all the links is larger (or smaller) than the corresponding QoS requirement, then this path is considered acceptable. Bandwidth is a typical example of the concave constraint. An additive constraint works as follows: in the case of a multi-link end-to-end path, the *sum* of all the QoS parameters along the path has to be less than the corresponding QoS requirement in order for this path to be acceptable. Delay is a typical example of the additive constraint. In Fig. 1, the path consists of 3 links: link 1, 2, and 3. Each of these links has a concave QoS parameter C_1, C_2, and C_3, respectively, and an additive QoS parameter A_1, A_2, and A_3, respectively. If a connection imposes QoS constraints C_0 and A_0, then the path is deemed acceptable if $\min\{C_1, C_2, C_3\} \geq C_0$, and $A_1+A_2+A_3 \leq A_0$.

One of the special characteristics of an additive constraint is that, from a per-link point of view, the QoS requirement of each link is related to the QoS behavior of all the other links in the same path. If we consider link 2 in Fig. 1 as an example, link 2 will be accepted if $A_2 \leq A_0 - (A_1 + A_3)$. Similarly, in an m-link path, a link among these m links, l_j $(j \leq m)$, is acceptable if

$$A_j \leq A_0 - \sum_{i=1, i \neq j}^{m} A_i.$$

Therefore, from the perspective of a single link, it cannot make the decision whether to accept or reject a connection purely based on its own additive link state metrics.

As we can see, the concave constraints have quite different properties from those of additive constraints; therefore, they have to be considered separately when designing a link state update scheme. Those aforementioned current link state update schemes (threshold, equal class, and exponential class updates) do not take the difference of these properties into consideration. In the next few sections, we will show how ROSE can cope with both concave and additive constraints better than the current link state update schemes.

III. False Routing

Ideally, when a connection request with certain QoS requirements is made to the network, the network's routing mechanism will accept this request and setup the connection if there are enough resources in the network to support the required QoS, and reject the request otherwise. However, in the real situation, since the routing mechanism does not always have the accurate link state information, it is unavoidable that some connections will be falsely accepted when the network actually cannot meet the QoS requirements; while some other connections will be falsely rejected when the network actually has enough resource to support the QoS requirements. In this paper, an instance of the first situation – a connection is falsely accepted – is referred to as a "false positive", and an instance of the second situation – a connection is falsely rejected – is referred to as a "false negative". Both false positives and false negatives constitute the definition of "false routing". In other words, we consider "false routing" has occurred as long as either a false positive or a false negative occurs.

False positives can jeopardize user's satisfaction since users are experiencing poor QoS in this situation. Meanwhile, false negatives can cause the under-utilization of network resources

[1]Preliminary results of ROSE have been presented in [16] and [17].

by rejecting the connections that should have been accepted. Therefore, both false positives and negatives are considered undesired situations. One can argue that one situation is more severe or, in other words, more costly, than the other. To reflect this concern, instead of simply gauge the performance of QoS routing by the probability of false routing, one should compare the "cost of false routing" for more realistic evaluation. A cost factor is used in ROSE, and therefore ROSE is not only capable of minimizing the occurrence of overall false routing, but also minimizing the overall cost of false routing. Throughout the rest of this paper, we will use the cost of false routing as the measure of the efficiency of various link state update schemes.

IV. Route-Oriented update SchEme (ROSE)

Here, we describe the new class-based link state update scheme, ROSE. The fundamental concept of ROSE is to utilize the statistical distribution of the user's QoS request and the network's QoS behavior in order to design an efficient class-based link state update scheme. The distribution of the user's QoS request can be obtained from the user profile (for example, x% of the connections requires y bps of bandwidth). The distribution of the network's QoS behavior can be derived from observing the operation history. Taking delay as an example, many reports have studied the delay measurements of various traffic types [18][19]. Reference [19] has proposed a method to measure the single-hop delay, represented as the frequency histogram of delay. Reference [20] directly indicates that the queue length of a bottlenecked link is likely to be Gaussian distributed as long as there is a large number of TCP sessions on this link at any given time. Since queuing delay is a major contributor of the end to end delay and possesses the most dynamic nature, the distribution of queue length can also be used to derive the pdf of single-hop delay. The subject of Internet measurements, which is a readily pursued research, is beyond the scope of this paper. In this paper, we thus assume the pdf's of the user's request and network's QoS behavior are known for the purpose of illustrating the ROSE algorithm. Consider a network composed of m links, denoted by the graph $G(V, E)$. We assume there is a routing device (either distributed or centralized) that makes the decision of whether to accept a connection request and finds the end-to-end paths that can provide the appropriate QoS to all accepted connections. The routing device makes the decision based on the link state information acquired via link state update. Generally, in a class-based link state update scheme, each link updates its QoS parameter by using a finite number of classes; here, let k be the number of classes. For a given QoS parameter, we further assume its value can only fall within a finite range (for example, the available bandwidth of a link can only be ranged from 0 up to the full link capacity). Therefore, with k classes, the ranges of the respective classes can be expressed as: $[B_{min}, B_1], [B_1, B_2], [B_2, B_3], \ldots, [B_{k-1}, B_{MAX}]$, where B_{min} and B_{MAX} are the minimum and maximum of the QoS parameter, and $B_1, B_2, \ldots, B_{k-1}$ are the boundaries of classes. To simplify the notations, we let $B_0 = B_{min}$ and $B_k = B_{MAX}$ throughout the rest of the paper. For each class, there is also a representative value which is "advertised" by the link to the routing device as if

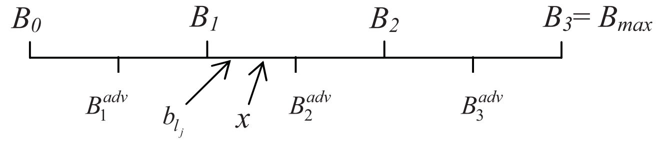

Fig. 2. Illustration of class boundaries and advertised values. In the concave case, a false positive occurs when $B_1 < b_{l_j} < x < B_2^{adv}$.

it is the exact value, denoted by $B_1^{adv}, B_2^{adv}, \ldots, B_k^{adv}$. For instance, if the available bandwidth of a link $l_j \in E$ falls in the range of $[B_1, B_2]$(class 2), then the link state update message will "advertise" that the available bandwidth of l_j is B_2^{adv}. In short, the links update their QoS status in a quantized manner. Fig. 2 illustrates the concept of class-based link state update. The routing device then makes the routing decision based on the advertised values from all the links. However, owing to quantization, false routing is inevitable. This can be illustrated by continuing the above example: When link l_j reports its available bandwidth as B_2^{adv}, the true value can be anywhere from B_1 to B_2. Therefore, if a connection attempting to utilize link j requests x amount of bandwidth and $B_1 < x < B_2^{adv}$, the routing device will accept this request. However, it is possible that the actual available bandwidth of link l_j is less than x but greater than B_1, and therefore incurring a false positive. On the other hand, if $B_2^{adv} < x < B_2$ and the actual available bandwidth of link l_j is greater than x but less than B_2, a false negative will occur (refer to Fig. 2). The goal of ROSE is to design the class boundaries and the advertised values intelligently to minimize the cost of false routing. Owing to the different properties of concave constraints and additive constraints as described in Section II, we have to consider them separately in the design of ROSE.

A. Concave QoS Constraints:

We start our analysis with a single concave QoS metric – bandwidth. When a connection requests x amount of bandwidth from link l_j, the connection will be accepted if $x < B^{adv}(l_j)$, where $B^{adv}(l_j)$ denotes the advertised available bandwidth of link l_j, and will be rejected otherwise. Assume that the actual available bandwidth of l_j, b_{l_j}, is within the range of class n ($1 \leq n \leq k$, k is the total number of classes), then $B^{adv}(l_j) = B_n^{adv}$. A false positive occurs when $x < B^{adv}(l_j)$ but $b_{l_j} < x$ (the actual bandwidth is less than the requested bandwidth). For this condition to hold, the following has to be true: $B_{n-1} < b_{l_j} < x < B^{adv}(l_j)$. Recall that we assume the statistical information of the user's QoS requirements and the network's QoS behavior are known, from which we can derive their corresponding probability density functions (pdf). Therefore, we can simply treat x and b_{l_j} as random variables. Let $q(x)$ be the pdf of the user's request x, and $p(b)$ be the pdf of the actual available bandwidth b_{l_j}, then we can write the probability of a false positive as:

$$Pr\{\textit{False Positive, class=n}\} = \int_{B_{n-1}}^{B_n^{adv}} \int_{b}^{B_n^{adv}} q(\tau)d\tau \cdot p(b)\,db. \tag{1}$$

Similarly, the probability of a false negative is:

$Pr\{\textit{False Negative, class=n}\}$

$$=\int_{B_n^{adv}}^{B_n}\int_{B_n^{adv}}^{b} q(\tau)d\tau \cdot p(b)\, db. \tag{2}$$

Equation (1) represents the situation of $B_{n-1} < b_{l_j} < x < B^{adv}(l_j)$, and (2) the situation of $B^{adv}(l_j) < x < b_{l_j} < B_n$. Note that (1) and (2) are *not* conditional probabilities; they describe the probability of false positive/negative AND the current class is n. Therefore, the overall probability of a false positive is

$$Pr\{False\ Positive\} = \sum_{n=1}^{k} Pr\{False\ Positive,\ class{=}n\}, \tag{3}$$

and the overall probability of a false negative is

$$Pr\{False\ Negative\} = \sum_{n=1}^{k} Pr\{False\ Negative,\ class{=}n\}\ . \tag{4}$$

Since the severity of a false positive and a negative might not be equal, let c_p be the cost of a false positive and c_n be that of a false negative; the total cost of false routing C can be written as:

$$\begin{aligned} C &= c_p \cdot Pr\{False\ Positive\} + c_n \cdot Pr\{False\ Negative\} \\ &= c_p \sum_{n=1}^{k} \int_{B_{n-1}}^{B_n^{adv}} \int_{b}^{B_n^{adv}} q(\tau)d\tau \cdot p(b)\, db \\ &\quad + c_n \sum_{n=1}^{k} \int_{B_n^{adv}}^{B_n} \int_{B_n^{adv}}^{b} q(\tau)d\tau \cdot p(b)\, db \end{aligned} \tag{5}$$

In order to minimize C with respect to B_n and B_n^{adv}, we need to find the solutions to the following equations:

$$\frac{\partial C}{\partial B_n} = 0 \Rightarrow c_n \int_{B_n^{adv}}^{B_n} q(\tau)d\tau - c_p \int_{B_n}^{B_{n+1}^{adv}} q(\tau)d\tau = 0 \tag{6}$$

$$\frac{\partial C}{\partial B_n^{adv}} = 0 \Rightarrow c_p \int_{B_{n-1}}^{B_n^{adv}} p(\tau)d\tau - c_n \int_{B_n^{adv}}^{B_n} p(\tau)d\tau = 0 \tag{7}$$

B. Additive Constraints:

In the analysis of additive constraints, we choose delay as our example for the rest of the paper. A unique property of additive constraints is that the decision of whether a link l_j can support the QoS requirement cannot be made based solely on this link's QoS measurement; it involves the QoS measurements of all other links along the path. Therefore, from the routing device's point of view, the decision of whether to select link l_j depends on whether

$$B^{adv}(l_j) < x - \sum_{i \in path, i \neq j} B^{adv}(l_i). \tag{8}$$

In other words, the decision is made based on whether the advertised delay of l_j is less than the user's request x subtracted by the sum of the advertised delays of all other links in the potential path. Again, since we assume the statistical information of x (request) and b_{l_j} (actual delay in link l_j) is available, x and $B^{adv}(l_j)$ can be treated as random variables, where the pdf of $B^{adv}(l_j)$ can be derived from the pdf of b_{l_j}. Then, the right half of (8) can be viewed as the sum of random variables. Let $S = x - \sum_{i \in path, i \neq j} B^{adv}(l_i)$, and $f_S(s)$ be the pdf of S. Essentially, S is the criterion of whether the connection will be accepted to utilize link l_j. Therefore, S will be referred to as the "accept/reject criterion" in this paper. Applying the Central Limit Theorem, $f_S(s)$ can be approximated by Gaussian distribution whose mean and variance can be derived from the pdf of x and $B^{adv}(l_j)$. Note that the mean and variance are affected by the number of hops in a connection. To simplify this problem, ROSE adopts the average hop count in a network to estimate $f_S(s)$. As we will show in our simulations, this simplified estimation still produces better performance for ROSE than equal-class and exponential-class link state updates.

Assume that the actual delay of link l_j falls in class n, i.e., its advertised delay $B^{adv}(l_j) = B_n^{adv}$. On the per-link basis, a false positive occurs when $B^{adv}(l_j) < S < b_{l_j} < B_n$, and a false negative occurs when $B_{n-1} < b_{l_j} < S < B^{adv}(l_j)$. Therefore, we can write the probability of a false positive and a false negative as:

$$\begin{aligned} &Pr\{False\ Positive,\ class{=}n\} \\ &\quad = \int_{B_n^{adv}}^{B_n} \int_{B_n^{adv}}^{b} f_S(s)ds \cdot p(b)\, db, \end{aligned} \tag{9}$$

$$\begin{aligned} &Pr\{False\ Negative,\ class{=}n\} \\ &\quad = \int_{B_{n-1}}^{B_n^{adv}} \int_{b}^{B_n^{adv}} f_S(s)ds \cdot p(b)\, db, \end{aligned} \tag{10}$$

where $p(b)$ is the pdf of the actual delay distribution of l_j.

From (9) and (10), we can follow the same procedure as in the analysis for concave constraints to obtain the overall cost of false routing:

$$\begin{aligned} C &= c_p \cdot Pr\{False\ Positive\} + c_n \cdot Pr\{False\ Negative\} \\ &= c_p \sum_{n=1}^{k} \int_{B_n^{adv}}^{B_n} \int_{B_n^{adv}}^{b} f_S(s)ds \cdot p(b)\, db \\ &\quad + c_n \sum_{n=1}^{k} \int_{B_{n-1}}^{B_n^{adv}} \int_{b}^{B_n^{adv}} f_S(s)ds \cdot p(b)\, db \end{aligned} \tag{11}$$

Again, to find B_n and B_n^{adv}(*n=1,… ,k*), we need to solve the following equations:

$$\frac{\partial C}{\partial B_n} = 0 \Rightarrow c_p \int_{B_n^{adv}}^{B_n} f_S(s)ds - c_n \int_{B_n}^{B_{n+1}^{adv}} f_S(s)ds = 0 \tag{12}$$

$$\frac{\partial C}{\partial B_n^{adv}} = 0 \Rightarrow c_n \int_{B_{n-1}}^{B_n^{adv}} p(\tau)d\tau - c_p \int_{B_n^{adv}}^{B_n} p(\tau)d\tau = 0 \tag{13}$$

Solving (6)-(7) and (12)-(13) requires a certain degree of computational complexity. However, the advantage of ROSE is that once the boundaries of the classes (B_n's) and their respective advertised values (B_n^{adv}'s) are solved, they can be simply plugged into each corresponding router so that the routers will perform link state update accordingly. In a network where the traffic pattern varies at different time of the day, the traffic pattern can be first categorized into different types for different time periods (such as peak-hour/off-peak-hour traffic, etc.), then each of them will have a separate set of B_n's and B_n^{adv}'s which will be in effect during its corresponding time period. As long as the traffic of the same type does not change drastically from day to day, (that is, say, every workday's traffic pattern between 9am to 11am is similar) we do not need to re-calculate the B_n's and B_n^{adv}'s. Therefore, when the network is in operation, aside from applying different B_n's and B_n^{adv}'s at different time periods of the day, ROSE will not incur additional computational overhead than equal-class or exponential-class link state updates.

V. Simulations

We evaluate the performance of ROSE by comparing it with the existing class-based update policies in [10]. For completeness, we briefly review the equal class based and exponential class based update policies.

Definition 1: Equal class based update policy [10] is characterized by a constant B which is used to partition the available bandwidth or delay operating region of a link into multiple equal size classes: *(0,B), (B, 2B), (2B, 3B),...,* etc. An update is triggered when the available bandwidth on an interface changes to a class that is different from the one at the time of the previous update.

Definition 2: Exponential class based update policy [15] is characterized by two constants B and *f (f>1)* which are used to define unequal size classes: *(0,B), (B, (f+1)B), ((f+1)B,* $(f^2+f+1)B),\ldots,$ etc. An update is triggered when a class boundary is crossed.

Concave Constraint: Bandwidth

The network topology used in the simulation is a 32-node network [15]. We adopt two performance indices for the purpose of comparison: the update rate (average number of updates in a unit time) and the false routing probability of connections, which are respectively defined below:

$$\text{Update rate}=\frac{\text{Total number of updates}}{(\text{Total simulation time})\cdot(\text{Number of links})},$$

and

$$\text{False routing probability} \\ =\frac{\text{number of falsely-routed connections}}{\text{number of connection requests}}.$$

The arrivals of connection requests are generated by a Poisson process with arrival rate $\lambda = 1$ and the duration of each connection is derived from the standard Pareto distribution with α=2.5 (the cumulative distribution of the standard Pareto distribution is $F(x) = 1 - (\beta/x)^{\alpha}$, where α is the shape parameter and β is the scale parameter). Hence, the average duration of a connection is $d = \alpha\beta/(\alpha-1)$ (the mean of the standard Pareto distribution). Upon the acceptance and the end of a connection, the available bandwidth is re-computed. The bandwidth requested by each connection is uniformly distributed in $[b_{min}, b_{max}]$, that is $q(x) \sim u(b_{\min}, b_{\max})$ in Eqs. (1) and (2) (do not confuse this with the actual available bandwidth which is distributed in $[0, C]$, where C is the link capacity.)

Without loss of generality, we assume the costs of a false positive and a negative are equal. Note that for a single class based link state update policy, the larger number of the classes the bandwidth is partitioned into, the more accurate the link state information is, implying the lower false blocking probability of connections, while the more sensitive it is to the fluctuation of the available bandwidth, thus resulting in a larger update rate. Hence, we can claim that policy 1 outperforms policy 2 if and only if, for any given number of classes used for policy 2, an appropriate number of classes can always be found for policy 1 such that it achieves better performance in terms of both the update rate and false routing probability of connections. By extensive simulations, we found that our proposed link state update policy outperforms the equal and exponential class based link state update policies for any given number of classes. In this paper, owing to the page limit, we only selectively present the simulation results of the cases that

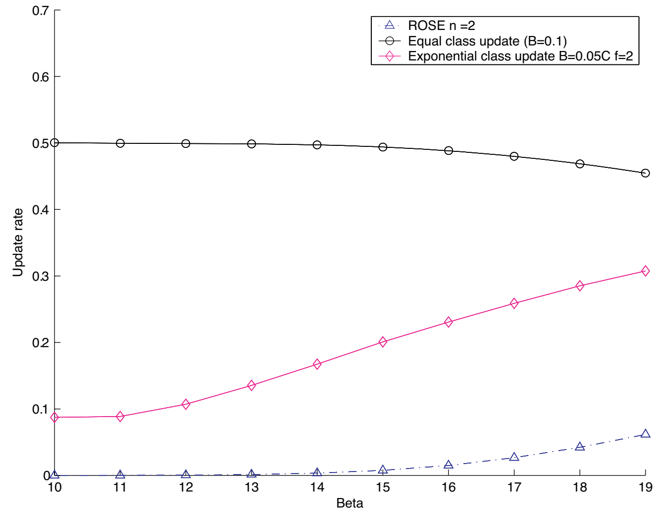

Fig. 3. Update rate when $[b_{min}, b_{max}]$ = [0, 0.05C].

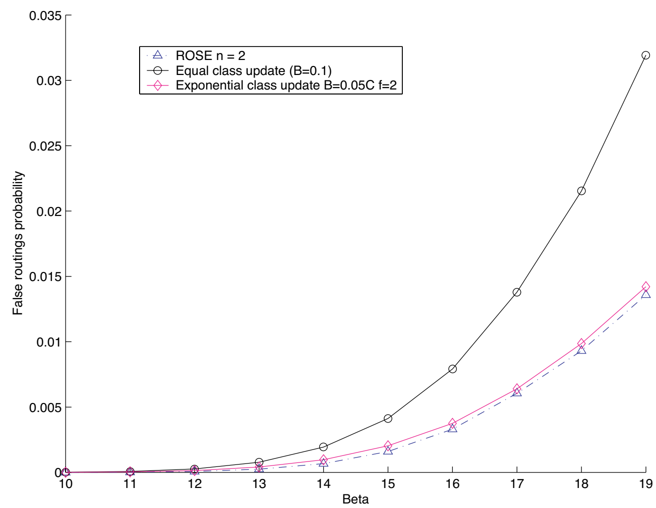

Fig. 4. False routing probability when $[b_{min}, b_{max}]$ = [0, 0.05C].

the numbers of classes of the equal class based update policy is 10 ($B = 0.1C$), and for the exponential class based update policy, $B = 0.05C$ and $f = 2$ (the number of classes is 5). In the two simulations, we set $[b_{min}, b_{max}]$ as [0, 0.05C] and [0.05C, 0.1C], and the number of classes of ROSE are 3 and 4, respectively.

As the first step of our proposed link state update policy, we compute the classes to partition the bandwidth. Since we assume the requested bandwidth is uniformly distributed in our simulations and the costs of false positive and negative are equal, (6) and (7) can be solved as:

$$B_n = b_{\min} + \frac{n\cdot(b_{\max}-b_{\min})}{k}$$

and

$$B_n^{adv} = \frac{(b_{\max}-b_{\min})}{2},$$

where k is the number of classes in ROSE.

Hence, the class based update policy adopted in our simulations is obtained. Figs. 3-6 illustrate our simulation results, in which *Beta* denotes the scale parameter β. In both simulations,

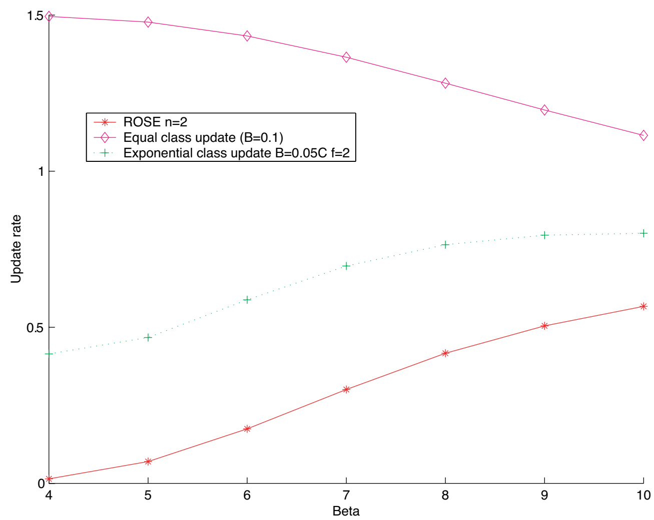

Fig. 5. Update rate WHEN $[b_{min}, b_{max}]$ = [0.05C, 0.10C].

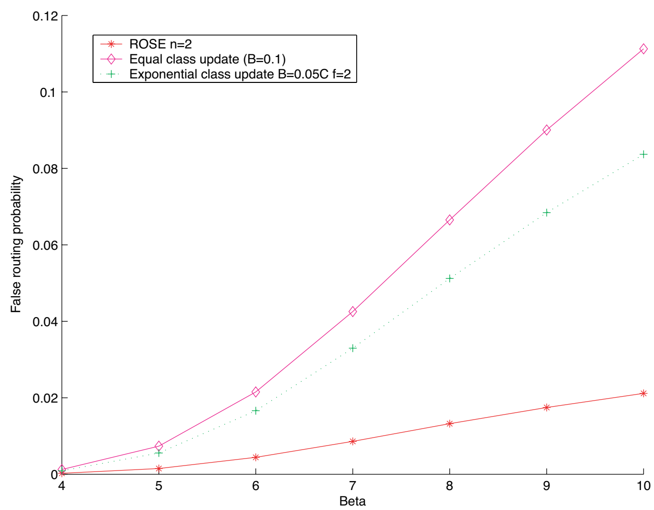

Fig. 6. False routing probability WHEN $[b_{min}, b_{max}]$ = [0.05C, 0.10C].

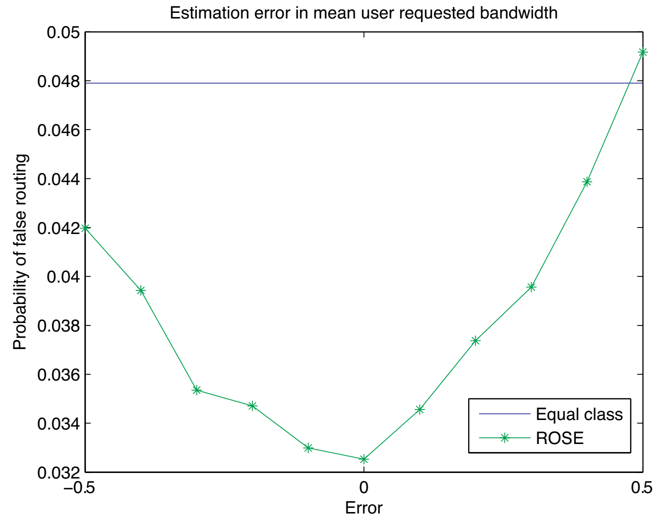

Fig. 7. False routing probability when there is error in measuring user's mean bandwidth request. Actual request pdf $q(x) \sim N\left(0.3C, 0.02C^2\right)$.

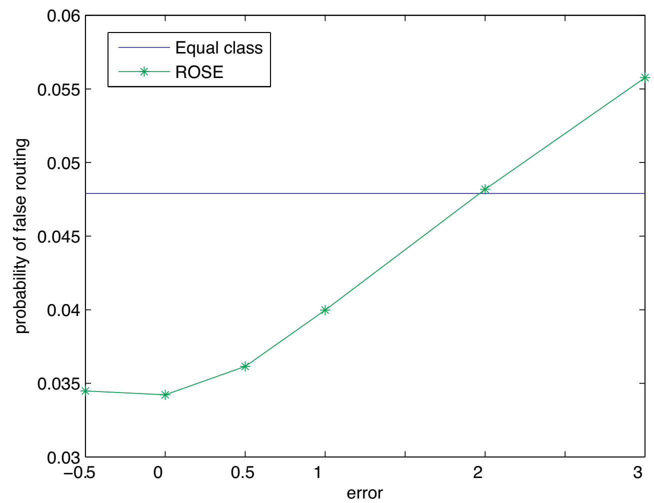

Fig. 8. False routing probability when there is error in measuring user's bandwidth request variance. Actual request pdf $q(x) \sim N\left(0.3C, 0.02C^2\right)$.

our proposed link state update policy achieves much better performance than others, i.e., our proposed link state update policy achieves lower false routing probabilities with lower update rates than others, implying that our proposed link state update is more practical than the equal and exponential class based link state update policies in terms of the update rate and false blocking probability of connections.

Concave Constraint with Error in pdf Estimation:

Since ROSE relies on the estimation of the pdf's of the user's request and the network's QoS behavior, it is important to examine the impact of erroneous estimation (in other words, fault tolerance). Here, we use bandwidth for illustrative purposes. In Fig. 7, error is introduced in measuring the mean of the user's bandwidth request: the actual distribution is assumed to be $q(x) \sim N\left(0.3C, 0.02C^2\right)$ while the estimated pdf is $\overline{q}(x) \sim N\left(0.3C * (1 + error), 0.02C^2\right)$. In Fig. 8, the error resides in measuring the variance of user's bandwidth request distribution. The incorrectly estimated pdf is $\overline{q}(x) \sim N\left(0.3C, 0.02C^2 * (1 + error)\right)$. For both experiments, the networks actual available bandwidth distribution is assumed to be exponentially distributed. The resulting probability of false routing is compared with that of the equal-class update. From these experiments, the ROSE algorithm exhibits a good degree of fault tolerance.

Additive Constraint: Delay

Let D_{MAX} be the maximum amount of delay a link can experience in the network (e.g., queue full). The accept/reject criterion S (recall that $S = x - \sum_{i \in path, i \neq j} B^{adv}(l_i))$ is simulated as normally distributed with mean $=0.3 \cdot D_{MAX}$, and variance $=3 \cdot D_{MAX}$. The actual delay distribution of D_{actual} is approximated as exponentially distributed in the simulation. One hundred thousand connection setup attempts were made, each time with a different value of accept/reject criterion S and a different D_{actual}. The class boundaries and their corresponding advertised delay B_n^{adv} were calculated according to (12) and (13). D_{MAX} is fixed at 1000 units.

Fig. 9 shows the results of the simulation when the number of classes varies from 3 to 12. As we can see, when the number of classes increases, the probability of false routing from either

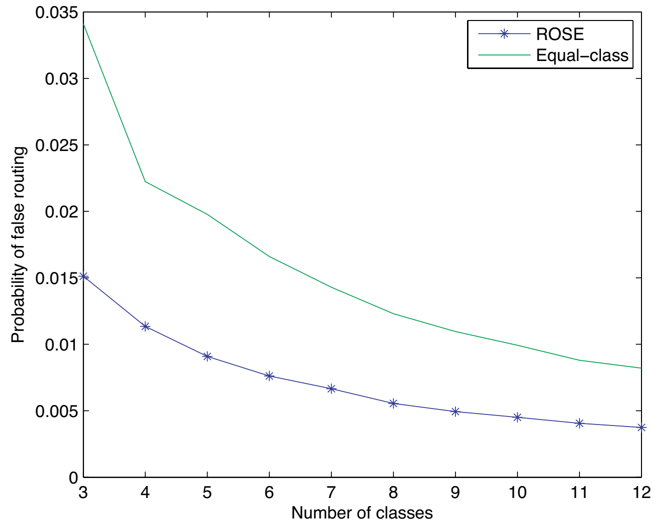

Fig. 9. Probability of false routing with varying number of classes. (ROSE vs. equal class.)

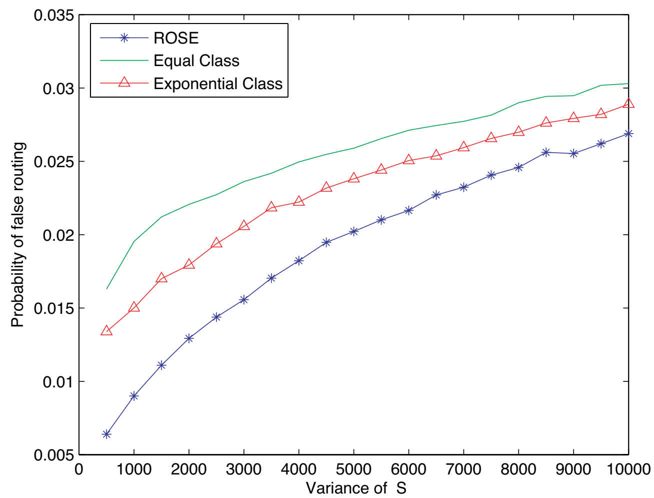

Fig. 10. Probability of false routing with varying variance of S. (ROSE versus exponential class and equal class.)

ROSE or equal-class updates decreases. This is due to the fact that the more classes, the more accurate link state information the network can obtain. However, regardless of the number of classes, ROSE always performs better than equal-class update, especially when the number of classes is small because equal-class update does not take the accept/reject criterion C into consideration.

Fig. 10 shows the results where the number of classes is fixed to 5 but the variance of S is varying from 1000 to 10000. From this figure, we can see that when the variance of accept/reject criterion S increases, the probability of false routing increases. However, ROSE still performs better than equal-class and exponential class updates, especially when the variance is low. Since ROSE takes the probability distribution of the accept/reject criterion into consideration, the lower variance means the accept/reject criterion is more predictable, hence yielding better performance for ROSE.

Fig. 11 compares the performance between ROSE and exponential-class update. Here, the number of classes is 5 and

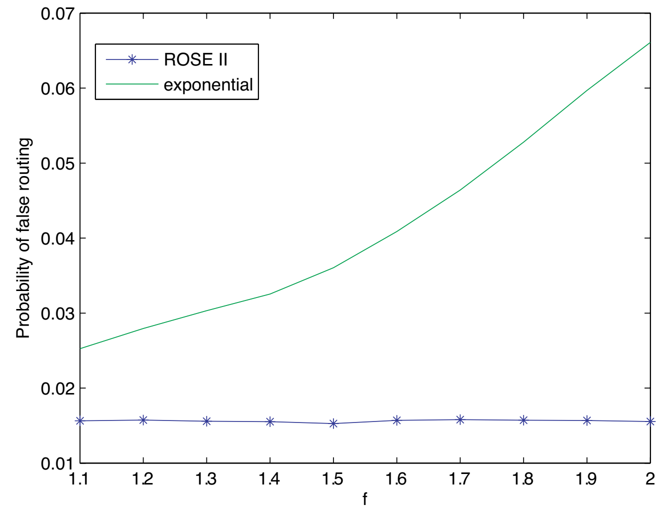

Fig. 11. Probability of false routing with varying f (ROSE vs. exponential class.

the variance of the accept/reject criterion is $3 \cdot D_{MAX}$. The factor f in exponential update varies from 1.1 to 2.0. We can see that the probability of false routing with ROSE remains almost constant because the change of f does not affect ROSE. However, the performance of exponential-class update decays slightly as the value of f increases. Exponential-class update can be viewed as a special case of ROSE in which all QoS parameters are exponentially distributed; in such case the ROSE algorithm would also yield class sizes (optimized) resembling those of exponential-class update. Nevertheless, the simulation result indicates that ROSE still performs better than exponential-class update even under exponentially distributed additive QoS parameters. This is because, for additive constraints, even if the QoS parameter of an individual link is exponentially distributed, the accept/reject criterion S is not. Therefore, the merit of ROSE is clearly revealed here.

Additive Constraints with various hop counts:

As we have previously pointed out, the pdf estimation of the accept/reject criterion S is based on the average hop count in the network. Obviously, ROSE serves well for the connections with the hop count equal to the average hop count. However, it is important to observe the impact to the other connections with different numbers of hops. For this purpose, we simulate a network in which the delay distribution is exponentially distributed with mean 8x10^3 units and variance 6.4x10^7 unit^2. The user's request is Gaussian distributed with mean 10^5 units and variance 10^8 unit^2. We assume the average hop count is 5, and therefore by applying the Central Limit Theorem, f_S, the pdf of S, can be approximated by Gaussian distribution with mean 6x10^3 units and variance of 4.2x10^8 unit^2.

To observe the effect of ROSE on the connections with various hop counts away from the average, we run simulations over the connections with hop counts from as low as 2 up to 8. The performance is compared with equal-class update and exponential class update, as presented in Fig. 12. From the result, we can see that ROSE still performs better than both exponential-class and equal-class updates for different hop counts. It is interesting to notice that the larger number of hops a connection traverses, the higher chance of false routing

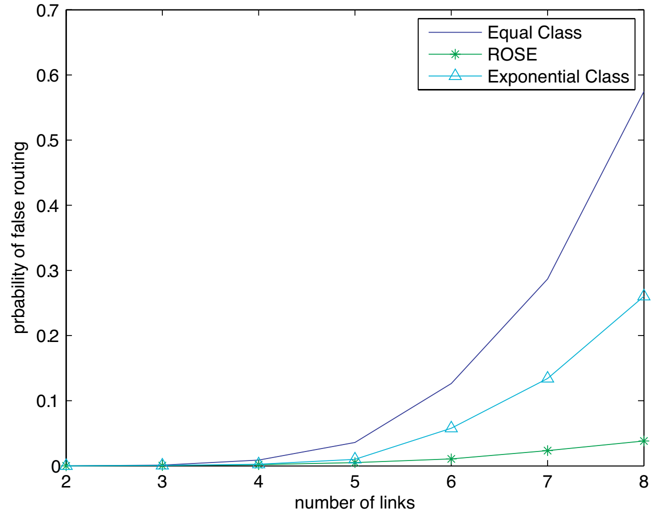

Fig. 12. Probability of false routing with various hop count. Average hop count =5. (ROSE versus exponential class and equal class.

it suffers. This is due to the fact that the inaccuracy of the link state information will accumulate from link to link in the case of additive constraints.

Summary

These simulation results demonstrate that ROSE yields lower probability of false routing than equal class update and exponential update in most of the scenarios. More importantly, ROSE also shows reasonable fault-tolerance even when the estimation of pdf is not accurate. In most of our simulations, the network QoS parameters are exponentially distributed while the user's request is normally distributed, but note that ROSE is applicable to different types of pdf's. The key here is to estimate the pdf's and solve Eqs. (12) (13); the more accurate the estimation, the better the performance of ROSE.

VI. Conclusion

In this paper, we have demonstrated that the statistical distribution of the user's QoS requirements and networks QoS measurements can be exploited to efficiently and effectively update link state information. We have proposed an efficient link state update policy, referred to as ROSE. Through theoretical analysis and extensive simulations, we have shown that ROSE greatly outperforms its contenders which do not incorporate the statistical information, i.e., ROSE achieves a much lower false routing probability and reduces the cost of false routing without significantly increasing the network overhead. Furthermore, ROSE can not only be applied to networks with various types of traffic and user requests, but is also capable of handling the dynamic nature of modern network traffic. ROSE can be the fundamental building block for QoS link state update in the next generation network.

References

[1] S. Chen and K. Nahrstedt, "An overview of quality of service routing for next-generation high-speed network: problems and solutions," *IEEE Network*, vol. 12, no. 6, pp. 64-79, Nov./Dec. 1998.

[2] G. Cheng and N. Ansari, "On multiple additively constraint path selection," *IEE Proc. Commun.*, vol. 149, no. 5, pp. 237-241, Oct. 2002.

[3] R. Guerin and A. Orda, "QoS based routing in networks with inaccurate information: theory and algorithms," in *Proc. IEEE INFOCOM'97*, Kobe, Japan, Apr. 1997, vol. 1, pp. 75-83.

[4] T. Korkmaz, M. Krunz, and S. Tragoudas, "An efficient algorithm for finding a path subject to two additive constraints," in *Proc. ACM SIGMETRICS'2000*, Santa Clara, CA, June 2000, pp. 318-327.

[5] L. Gang and K. G. Ramakrishnan, "A*prune: an algorithm for finding k shortest paths subject to multiple constraints," in *Proc. IEEE INFOCOM'2001*, Anchorage, AK, Apr. 2001, vol. 2, pp. 743-749.

[6] S. Chen and K. Nahrstedt, "Distributed QoS routing with imprecise state information," in *Proc. 7th Intl. Conf. Computer Communications and Networks*, Lafayette, LA, Oct. 1998, pp. 614-621.

[7] Z. Wang and J. Crowcroft, "Quality of service routing for supporting multimedia applications," *IEEE J. Select. Areas Commun.*, vol. 14, no. 7, pp. 1228-1234, Sept. 1996.

[8] J. Moy, "OSPF version 2," RFC2328, IETF, 1998.

[9] S. Sengupta, D. Saha, and S. Chaudhuri, "Analysis of enhanced OSPF for routing lightpath in optical mesh networks," in *Proc. IEEE ICC'02*, vol. 5, pp. 2865-2869, 2002.

[10] G. Apostolopoulos, R. Guerin, S Kamat, and S. Tripathi, "Quality-of-service based routing: a performance perspective," in *Proc. ACM SIGCOMM 1998*, Vancouver, BC, Canada, Aug. 1998, vol. 28, pp. 17-28.

[11] A. Shaikh, J. Rexford, and K. G. Shin, "Evaluating the impact of stale link state on quality-of-service routing," *IEEE/ACM Trans. Networking*, vol. 9, no. 2, pp. 162-176, Apr. 2001.

[12] Q. Ma and P. Steenkiste, "Quality-of-service routing for traffic with performance guarantees," in *Proc. IFIP Int. Workshop Quality of Service*, New York, May 1997, pp. 115-126.

[13] L. Breslau, D. Estrin, and L. Zhang, "A simulation study of adaptive source routing in integrated service networks," Computer Science Department, University of Southern California, Tech. Rep. 93-551, 1993.

[14] M. Peyravian and R. Onvural. "Algorithm for efficient generation of link-state updates in ATM networks," *Computer Networks and ISDN Systems*, vol. 29, pp. 237.247, 1997.

[15] X. Li, L. K. Shan, W. Jun, and N. Klara, "QoS extension to BGP," in *Proc. IEEE Intl. Conf. Network Protocols (ICNP'02)*, Paris, France, Nov. 2002, pp. 100-109.

[16] G. Cheng and N. Ansari, "ROSE: a novel link state information update scheme for QoS routing," in *Proc. 2005 IEEE Workshop on High Performance Switching and Routing*, Hong Kong, May, 2005. pp. 24-28.

[17] N. Wang, G. Cheng, and N. Ansari. "ROSE II: the case of additive metrics for updating additive link state information," in *Proc. ICC 2006*, Istanbul, Turkey, May 2006.

[18] J. Ding, M. Kirkpatrick, and E. H.-M. Sha. "QoS measures and implementations based on various models for real-time communications," in *Proc. 3rd IEEE Symposium on Application-Specific Systems and Software Engineering Technology 2000*, March 2000, pp. 125-129.

[19] L. Angrisani, G. Ventre, L. Peluso, and A. Tedesco. "Measurement of processing and queuing delays introduced by an open-source router in a single-hop network," *IEEE Trans. Instrumentation and Measurement*, vol. 55, no. 4, pp. 1065-1076, Aug. 2006.

[20] G. Appenzeller, I. Keslassy, and N. McKeown. "Sizing router buffers," *Computer Commun. Rev.*, vol. 34, no. 4, pp. 281-292, 2004.

Nirwan Ansari (S'78-M83-SM'94) received the B.S.E.E. (summa cum laude) from the New Jersey Institute of Technology (NJIT), Newark, in 1982, the M.S.E.E. degree from University of Michigan, Ann Arbor, in 1983, and the Ph.D. degree from Purdue University, West Lafayette, IN, in 1988. He joined NJIT's Department of Electrical and Computer Engineering as an Assistant Professor in 1988, and has been a Full Professor since 1997. He has also assumed various administrative positions. He authored Computational Intelligence for Optimization (Springer, 1997, translated into Chinese in 2000) with E.S.H. Hou, and edited Neural Networks in Telecommunications (Springer, 1994) with B. Yuhas. His current research focuses on various aspects of broadband networks and multimedia communications. He has also contributed over 300 technical papers including over 100 refereed journal/magazine articles.

He is a Senior Technical Editor of the *IEEE Communications Magazine*, and also serves on the editorial board of *Computer Communications*, the *ETRI Journal*, and the *Journal of Computing and Information Technology*.

He was the founding general chair of the First IEEE International Conference on Information Technology: Research and Education (ITRE2003), was instrumental, while serving as its Chapter Chair, in rejuvenating the North Jersey Chapter of the IEEE Communications Society which received the 1996 Chapter of the Year Award and a 2003 Chapter Achievement Award, served as Chair of the IEEE North Jersey Section and in the IEEE Region 1 Board of Governors during 2001-2002, and has been serving in various IEEE committees such as Chair of IEEE COMSOC Technical Committee on Ad Hoc and Sensor Networks, and (TPC) Chair/Vice-chair of several conferences/symposia. His awards and recognitions include the NJIT Excellence Teaching Award in Graduate Instruction (1998), IEEE Region 1 Award (1999), an IEEE Leadership Award (2007, from IEEE Princeton/Central Jersey Section), and designation as an IEEE Communications Society Distinguished Lecturer.

Gang Cheng received B.S. in Information Engineering from Beijing University of Posts and Telecommunications (BUPT) in 1997. He joined Lucent Technologies after he obtained his M.E. in Information and Signal Processing from BUPT in 2000. Between January 2001 and May 2005, he was with the New Jersey Institute of Technology. His research interests include Internet routing protocols and service architectures, information theory based network optimization and protocol design, and modeling and performance evaluation of computer and communication systems. He received the Ph.D. degree from NJIT in May 2005, and was the recipient of the Hashimoto Prize, which is awarded annually to the "best" NJIT doctoral graduate in Electrical and Computer Engineering. He joined VPIsystems Corp. in Jan 2005, focusing on the network planning optimization algorithm design and development. Since May 2006, he has been with EMC, working on the root-cause analysis network management design and development.

Nan Wang received the B.S.E.E. degree from National Cheng Kung University, Tainan, Taiwan in 1995, and the M.S.E.E. degree from the New Jersey Institute of Technology, Newark, New Jersey in 1998. He has worked for Lucent Technologies as a network engineer in the area of ATM networks, and Comcast in the VoIP. He is currently working towards the Ph.D. degree in electrical engineering at NJIT, and his main research interests include QoS routing, link state updating, and performance evaluation.