

On QoS Routing and Path Establishment in the Presence of Imprecise State Information

Xin Yuan and Zhenhai Duan

Abstract: In large networks, maintaining precise global network state information is impossible. The imprecise network state information has significant impacts on Quality-of-Service (QoS) routing and path establishment. In this paper, we develop analytical models for investigating the QoS routing schemes that are designed to tolerate imprecise network state information, and study the performance of various QoS routing schemes, including randomized routing, multi-path routing, and probability based routing, using both the models and simulations. We further propose a new mechanism, called *local rerouting*, that improves the effectiveness of QoS path establishment in the presence of imprecise global network state information. Local rerouting is orthogonal to existing schemes for dealing with imprecise state information and can be used in conjunction with them to alleviate the impacts of the imprecise information. The results of our performance study indicate that local rerouting is robust and effective.

Index Terms: Imprecise network state, local re-routing, Quality-of-Service (QoS), routing.

I. INTRODUCTION

In a Quality of Service (QoS) enabled network such as an MPLS network, the path establishment mechanism is responsible for setting up paths that satisfy QoS constraints. The path establishment mechanism contains two components: QoS routing that computes the paths that satisfy QoS constraints and resource reservation that establishes QoS paths using some protocol such as RSVP [1]. In this paper, we consider the problem of dynamically establishing QoS paths in the presence of imprecise network state information. The motivation for this problem is that in large networks, it is almost impossible to maintain precise network state information. 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 [2]. The imprecise global state information can significantly affect the effectiveness of QoS routing (and hence path establishment) [3].

We assume that the global network state information is maintained by a link state algorithm and the QoS metric is the bandwidth. The terms *flow* and *connection* will be used interchangeably in this paper. Traditional QoS routing algorithms select feasible paths based on the link state (link resource availability) information. We will call such routing schemes *state based routing*. As shown in [3], the performance of a state based routing

algorithm can be significantly affected by the imprecise network state information. To overcome this problem, various routing schemes that explicitly take the fact that the link state information is imprecise into account were developed. Such schemes can be broadly classified into three categories: *Probability based routing* where the link state parameters are modeled as random variables and the path that has the highest probability to satisfy the constraints is selected [2], [4]–[9], *multi-path routing* where multiple paths are probed for each connection request [10]–[12], and *randomized routing* where a path is randomly selected from a set of feasible paths that are computed based on the imprecise network state information [4], [13].

In this paper, we develop analytical models for investigating the QoS routing algorithms and use these models to study the performance of the algorithms with imprecise state information. Our investigation, which is the first time that analytical models are used to study these schemes, yields new insights about how the routing schemes behave. Simulations on various Internet-like topologies confirm the findings in the analytical study. The major conclusions of our study include:

- The potential achievable performance of probability based routing is much lower than the potential achievable performance of state based routing. Thus, in a sense, probability based routing is inherently inferior to state based routing although probability based routing can be effective when the network state information is very imprecise.
- Randomized routing is problematic in dealing with imprecise network state information.
- Multi-path routing is always effective in dealing with imprecise network state information.

While multi-path routing is robust in tolerating imprecise network state information, existing multi-path routing schemes either probe multiple paths concurrently [10]–[12], which introduces much higher overheads than their single path counterparts, or probe multiple paths sequentially as in ATM PNNI [14], which incurs a long path setup time. We propose a new scheme, called *local rerouting* (LR), that allows multiple paths to be probed while avoiding the problems in the current multi-path routing schemes. The idea is to use local information to compute by-pass paths for each link and to perform dynamic rerouting in case when a link in the computed feasible path cannot satisfy the QoS requirement during the resource reservation process due to the stale link state information. The information used in LR is local to a router because only the link state information within a few hops from the router is needed (the number of hops is a parameter in LR).

LR has several desirable features. First, LR is orthogonal to existing mechanisms that are designed to tolerate imprecise state information. LR can be combined with them to further improve performance. Second, LR can be deployed incrementally. It

Manuscript received May 8, 2007.

This work was supported in part by National Science Foundation grants ANI-0106706 and CCF-0551555.

The authors are with the Department of Computer Science, Florida State University, Tallahassee, FL 32306, email: {xyuan, duan}@cs.fsu.edu.

only requires the support of a small region of routers to work. Third, the by-pass paths in LR are computed using local information and are not subject to the effects of imprecise global network state information. The results of our performance study indicate that LR is a robust and efficient scheme: It improves performance while only introducing minor overheads.

The rest of the paper is organized as follows. Section II describes the related work. In Section III, we present our analytical models and study the performance of QoS routing algorithms using both the models and simulations. In Section IV, we present and evaluate our proposed new mechanism, local rerouting. Finally, Section V concludes the paper.

II. RELATED WORK

The problem caused by imprecise global network state information has caught much attention recently. It was found in [3], [4] that imprecise network state information can greatly degrade the performance of QoS routing. To overcome this problem, better link state update mechanisms including the threshold and class based schemes were proposed [3], [4] and a number of QoS routing schemes that explicitly consider the imprecise network state information were developed [2], [4]–[8], [10]–[13]. These algorithms can be classified into probability based routing [2], [4]–[8], multi-path routing [10]–[12], and randomized routing [4], [13]. Guerin and Orda [2] proved a number of important theoretical results on routing in networks with imprecise state information and proposed QoS routing schemes based on probability. In [4], the probability based QoS routing scheme in [2] was materialized to be the safety-based routing which infers the probability information from the availability information based on the link state update policy. Work in [5] and [8] proposed to explicitly maintain the link bandwidth probability distribution information based on empirical statistics. An empirical study of QoS routing algorithms that tolerate imprecise network state information was present in [15]. In [6] and [9], techniques were developed for QoS routing with multiple QoS metrics under imprecise state information. All the existing work relies on simulation to study the impacts of the imprecise state information and to evaluate the QoS routing algorithms. We develop analytical models for studying the impacts of the imprecise state information and propose a new mechanism to deal with the imprecise network state information.

III. ANALYTICAL MODELS

For a QoS routing algorithm to effectively tolerate the imprecise state information, it must be able to explore the multiple paths between the source and the destination and identify a feasible path among these paths with the imprecise information. The ability of a QoS routing algorithm to explore the multiple paths between the source and the destination in the fish network shown in Fig. 1 gives a good indication about the capability of the algorithm in exploring multiple paths between two nodes in a general network. We develop analytical models that give the blocking probability of various QoS routing algorithms on the fish topology. These models allow us to quantify the impacts of the imprecise state information and to quantitatively compare

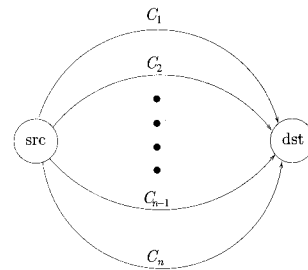


Fig. 1. The fish topology.

the QoS routing algorithms.

The analytical models make the following common assumptions. Additional assumptions for a particular model will be specified as the model is described. In the fish topology, there are n paths between the source and the destination. The capacity of path i is C_i , $1 \leq i \leq n$. Let the remaining bandwidth at path i be r_i , the network state can be represented as $\vec{N} = (r_1, r_2, \dots, r_n)$. For a network state, the paths can be ordered by their remaining bandwidths and their path ranks. More specifically, for a network state $\vec{N} = (r_1, r_2, \dots, r_n)$, path i has a larger remaining bandwidth than path j if and only if (1) $r_i > r_j$ or (2) $r_i = r_j$ and $i < j$. Using this ordering, for a given network state, the path with the largest remaining bandwidth in the system can be identified. Similarly, k paths, $k \leq n$, with the largest remaining bandwidths can also be identified.

The flows arrive at the source node according to a Poisson process with a rate λ . The connection holding time is exponentially distributed with a mean of $\frac{1}{\mu}$ seconds. Each flow requires 1 unit of bandwidth. The link state update interval is exponentially distributed with a mean of $\frac{1}{\delta}$ seconds. We will use the following notations in the analytical models:

- $C_{sum} = C_1 + C_2 + \dots + C_n$: the capacity of all paths.
- $C_{max} = \max\{C_1, C_2, \dots, C_n\}$: the largest single path capacity.
- $E(\rho, C) = \frac{\rho^C / C!}{\sum_{c=0}^C \rho^c / c!}$: the celebrated Erlang loss formula, which gives the blocking probability for a system with a capacity of C units of bandwidth and an offer load of $\rho = \frac{\lambda}{\mu}$.
- $m! = m \times (m-1) \times \dots \times 1$: the factorial function.
- $\binom{i}{j} = \frac{i!}{j!(i-j)!}$: the number of j combinations of an i -set.

Widest-Shortest Path Algorithm

The widest-shortest path algorithm (*wsr*) [16] is a state based routing algorithm that does not incorporate any mechanism to tolerate imprecise state information. It selects the shortest path that can satisfy a connection request for the request. When there are multiple such paths, the one that has the largest remaining bandwidth is selected. When using the widest-shortest path algorithm on the fish topology, the algorithm always selects the path that has the largest remaining bandwidth for a connection request. The state of the system can be described by the network state $\vec{s} = (r_1, r_2, \dots, r_n)$, where r_i , $1 \leq i \leq n$, is the remaining bandwidth on path i . When a flow arrives, *wsr* selects the path with the largest bandwidth for the flow. The stochastic process corresponding to this state vector is a finite-

state continuous-time Markov process with the state space: $S = \{(r_1, r_2, \dots, r_n) : 0 \leq r_1 \leq C_1, \dots, 0 \leq r_n \leq C_n\}$. The number of states in S is equal to $(C_1+1) \cdot (C_2+1) \cdot \dots \cdot (C_n+1)$. The state transition rate, $q((a_1, a_2, \dots, a_n), (b_1, b_2, \dots, b_n))$, from state (a_1, a_2, \dots, a_n) to another state (b_1, b_2, \dots, b_n) is

$$q((a_1, a_2, \dots, a_n), (b_1, b_2, \dots, b_n)) = \begin{cases} (C_i - a_i) \times \mu, & \text{if } (b_i = a_i + 1) \text{ and} \\ & (a_j = b_j) \text{ for all } j \neq i; \\ \lambda, & \text{if } (b_i = a_i - 1) \text{ and } (a_j = b_j) \text{ for all } j \neq i \\ & \text{and path } i \text{ has the largest remaining} \\ & \text{bandwidth in state } (a_1, a_2, \dots, a_n); \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

There are two types of transitions from state (a_1, a_2, \dots, a_n) , one for flow departures and one for flow arrivals. When one current flow on path i , $1 \leq i \leq n$, leaves the system, state $(a_1, \dots, a_{i-1}, a_i, a_{i+1}, \dots, a_n)$ transits to state $(a_1, \dots, a_{i-1}, a_i + 1, a_{i+1}, \dots, a_n)$ with a rate of $(C_i - a_i) \times \mu$ since there are $(C_i - a_i)$ active flows on path i , each having a departure rate of μ . When a new flow arrives, it is routed on the path with the largest bandwidth. Let path i be the path that has the largest remaining bandwidth in state $(a_1, \dots, a_{i-1}, a_i, a_{i+1}, \dots, a_n)$. When a flow arrives, state $(a_1, \dots, a_{i-1}, a_i, a_{i+1}, \dots, a_n)$ transits to state $(a_1, \dots, a_{i-1}, a_i - 1, a_{i+1}, \dots, a_n)$ with a rate of λ . Using transition formula (1), it is straightforward to derive the infinitesimal generator P for this Markov process. The equilibrium probabilities $\pi = (\pi(\vec{s}), \vec{s} \in S)$ are the solutions to $\pi P = 0$ along with $\sum_{\vec{s} \in S} \pi(\vec{s}) = 1$.

It can be shown that this Markov process has close form solutions for the equilibrium probabilities. In particular, $\pi((0, 0, \dots, 0)) = (\rho^{C_{sum}} / C_{sum}!) / (\sum_{c=0}^{C_{sum}} \rho^c / c!)$. Here, $\rho = \lambda / \mu$ and $C_{sum} = C_1 + C_2 + \dots + C_n$. Using wsr , a flow is blocked only when the network is in the state $(0, 0, \dots, 0)$. Thus, the blocking probability is as following:

$$B = \pi(0, 0, \dots, 0) = \frac{\rho^{C_{sum}} / C_{sum}!}{\sum_{c=0}^C \rho^c / c!}.$$

Notice that this result is equal to Erlang's loss formula for a single link with capacity C_{sum} , $E(\frac{\lambda}{\mu}, C_{sum})$. This indicates that with precise state information, the widest shortest routing algorithm can explore all paths between the source and the destination. For the special case when all paths have the same capacity Φ , the blocking probability is $E(\frac{\lambda}{\mu}, n\Phi)$.

Next, let us consider the extreme case when the link state update interval is close to infinite. Since the link state is never updated, wsr always selects the path with the largest capacity (instead of the one with the largest remaining bandwidth) for each connection request. The blocking probability is thus $E(\frac{\lambda}{\mu}, C_{max})$, where $C_{max} = \max\{C_1, C_2, \dots, C_n\}$. For the special case when all paths have the same capacity Φ , the blocking probability is $E(\frac{\lambda}{\mu}, \Phi)$.

When the link state update interval is small, wsr will use accurate state information for path computation and have a blocking probability close to $E(\frac{\lambda}{\mu}, C_{sum})$. When the link state update interval is large, wsr will use inaccurate state information

for path computation and have a blocking probability close to $E(\frac{\lambda}{\mu}, C_{max})$. In general, the blocking probability of wsr falls between these two bounds.

Let us consider the general case when the link state update interval is exponentially distributed with a mean of $\frac{1}{\delta}$ seconds (and thus, the global network state is imprecise). The system can also be modeled as a continuous-time Markov process. The state of the system is described by the vector

$$\vec{s} = (index, (r_1, \dots, r_n)).$$

The system state consists of two parts: The current network state (r_1, r_2, \dots, r_n) and the *index* of the path that has the largest bandwidth in the previous link state update. Adding *index* enables the state transition under the imprecise network state information to be modeled: All flows arrived are routed to path *index* since this path is considered by wsr as the best path. A new path is selected only when the link state is updated

$$S = \{(index, (r_1, r_2, \dots, r_n))\}$$

where $1 \leq index \leq n$ and $0 \leq r_1 \leq C_1, \dots, 0 \leq r_n \leq C_n$.

The number of states in S is equal to $n \prod_{k=1}^n (C_k + 1)$. The state transition rate from state $(i_a, (a_1, a_2, \dots, a_n))$ to another state $(i_b, (b_1, b_2, \dots, b_n))$ is

$$q((i_a, (a_1, a_2, \dots, a_n)), (i_b, (b_1, b_2, \dots, b_n))) = \begin{cases} (C_i - a_i) \times \mu, & \text{if } (i_a = i_b) \text{ and } (b_i = a_i + 1) \\ & \text{and } (a_j = b_j) \text{ for all } j \neq i; \\ \lambda, & \text{if } (i_a = i_b) \text{ and } (b_{i_b} = a_{i_a} - 1) \\ & \text{and } (a_j = b_j) \text{ for all } j \neq i_a; \\ \delta, & \text{if } (i_a \neq i_b) \text{ and } (a_j = b_j) \text{ for all } 1 \leq j \leq n \\ & \text{and path } i_b \text{ has the largest bandwidth} \\ & \text{in state } (a_1, a_2, \dots, a_n); \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

There are three types of transitions from $(i_a, (a_1, a_2, \dots, a_n))$ state, one for flow departures, one for flow arrivals, and one for link state updates. The first type of transitions corresponds to the case when one current flow on path i leaves the system. In this case, for each $1 \leq i \leq n$, the transition rate from state $(i_a, (a_1, \dots, a_{i-1}, a_i, a_{i+1}, \dots, a_n))$ to state $(i_a, (a_1, \dots, a_{i-1}, a_i + 1, a_{i+1}, \dots, a_n))$ is $(C_i - a_i) \cdot \mu$ since there are $(C_i - a_i)$ active flows on path i . The second type corresponds to the case when a new connection arrives and is routed on path i_a . In this case, the transition rate from state $(i_a, (a_1, \dots, a_{i_a-1}, a_{i_a}, a_{i_a+1}, \dots, a_n))$ to state $(i_a, (a_1, \dots, a_{i_a-1}, a_{i_a} - 1, a_{i_a+1}, \dots, a_n))$ is the connection arrival rate λ . The third type corresponds to a link state update, where a path with the largest remaining bandwidth is computed. Let path i_b be the path with the largest remaining bandwidth in state $(i_a, (a_1, \dots, a_n))$. The transition rate from state $(i_a, (a_1, \dots, a_n))$ to state $(i_b, (a_1, \dots, a_n))$ is δ . Using transition formula (2), it is straightforward to derive the infinitesimal generator for this Markov process and the equilibrium probabilities $\pi = (\pi(\vec{s}), \vec{s} \in S)$ can be computed in the similar way as those in the wsr with precise state information. Among all the states in the state space, the states that can result in

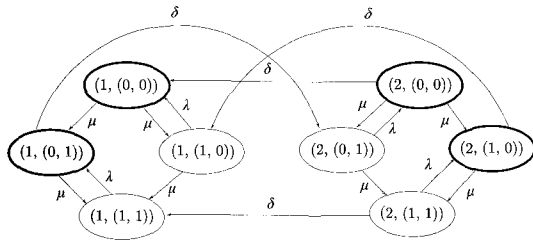


Fig. 2. The Markov chain for the case when $n = 2$ and $C_1 = C_2 = 1$.

blocking are the ones that do not have bandwidth on the designated path. The set of states, BS , that can result in blocking is $BS = \{(i, (a_1, \dots, a_n)) : (i, (a_1, \dots, a_n)) \in S \text{ and } a_i = 0\}$. The blocking probability is equal to

$$B = \sum_{(i, (a_1, \dots, a_n)) \in BS} \pi(i, (a_1, \dots, a_n)).$$

An example Markov chain for the case when $n = 2$ and $C_1 = C_2 = 1$ is shown in Fig. 2. In the figure, the states with darker borders are the states that will cause blocking. It is unclear whether the Markov model for the wsr with imprecise state information has close form solutions. However, numeric results can be obtained from this model to study this scheme.

Probability Based Routing

Different probability based schemes differ in how the probability information is obtained and maintained. They all use the same routing algorithm, the *shortest-safest routing* [4], for path computation. The *shortest-safest routing* algorithm first determines all paths that have the largest probability to satisfy a connection request, and then selects the shortest one for the connection request. Since precise probability information can usually be obtained, we will assume that precise probability information is available.

Using the probability based routing, when a path has a lower blocking probability, more flows will be routed through this path. Thus, on the fish topology, flows will be distributed among all paths such that all paths have the same blocking probability. Let α_i be the portion of all flows routed to path i , the blocking probability of path i is $E(\alpha_i \frac{\lambda}{\mu}, C_i)$. With probability based routing, the flow distribution can be obtained by solving equations:

$$E(\alpha_1 \frac{\lambda}{\mu}, C_1) = E(\alpha_2 \frac{\lambda}{\mu}, C_2) = \dots = E(\alpha_n \frac{\lambda}{\mu}, C_n). \quad (3)$$

along with $\alpha_1 + \alpha_2 + \dots + \alpha_n = 1$.

The iterative method can be used to obtain the solutions. Once the flow distributed is known (the values of α_i , $1 \leq i \leq n$, are known), the blocking probability of the probability based method is $E(\alpha_1 \frac{\lambda}{\mu}, C_1)$. For the special case when all paths have the same capacity Φ , the flows will be evenly distributed among all the n paths and the blocking probability is $E(\frac{\lambda}{\mu} \frac{1}{n}, \Phi)$. This is the highest performance that a probability based routing can achieve since we assume that the probability information is precise. Notice that using state based routing, the potential achievable blocking rate is $E(\frac{\lambda}{\mu}, n\Phi)$, which is much smaller than $E(\frac{\lambda}{\mu} \frac{1}{n}, \Phi)$, especially when n is large.

Multi-Path Routing

Multi-path routing [10], [12] assumes that some single path QoS routing algorithm is used to compute a set of feasible paths. For example, multi-path routing can repeatedly use wsr to compute the set of feasible paths. It then probes all the paths to determine a feasible path for the connection request. The model for multi-path routing assumes that wsr is repeated k times to find the set of k feasible paths to be probed, where k is a parameter for the algorithm. Thus, every time when the link state is updated, k paths that have the largest bandwidths in the current state are selected to be probed for determining feasible paths until the next link state update. When multiple feasible paths among the k paths can satisfy a connection request, the one with the largest remaining bandwidth is selected for the request. For multi-path routing, the state of the system is described by the vector

$$\vec{s} = (kindex, (r_1, \dots, r_n)).$$

In addition to the current network state (r_1, \dots, r_n) , the k selected paths must be recorded in $kindex$ in the system state. $kindex$ is some kind of encoding that records the k selected paths. Since there are $\binom{n}{k}$ number of ways to select k paths out of n total paths, the potential value for $kindex$ can be from 1 to $\binom{n}{k}$. By decoding $kindex$, k indices, i_1, i_2, \dots, i_k , for the k selected paths, can be derived. The state space is

$$S = \{(kindex, (r_1, r_2, \dots, r_n))\}$$

where $1 \leq kindex \leq \binom{n}{k}$ and $0 \leq r_1 \leq C_1, \dots, 0 \leq r_n \leq C_n$.

The number of states in S is equal to $\binom{n}{k} \prod_{i=1}^n (C_i + 1)$. Consider the transition rate from state $(ki_a, (a_1, a_2, \dots, a_n))$ to state $(ki_b, (b_1, b_2, \dots, b_n))$. Let decoding ki_a yield indices i_1, i_2, \dots, i_k . We will use the notation $max(ki_a)$ to represent the index in i_1, i_2, \dots, i_k that has the largest remaining bandwidth. The state transition rate from state $(ki_a, (a_1, a_2, \dots, a_n))$ to state $(ki_b, (b_1, b_2, \dots, b_n))$ is

$$q((ki_a, (a_1, a_2, \dots, a_n)), (ki_b, (b_1, b_2, \dots, b_n))) = \begin{cases} (C_i - a_i) \times \mu, & \text{if } (ki_a = ki_b) \text{ and } (b_i = a_i + 1) \\ & \text{and } (a_j = b_j) \text{ for all } j \neq i; \\ \lambda, & \text{if } (ki_a = ki_b) \text{ and } (b_{max(ki_a)} = a_{max(ki_a)} - 1) \\ & \text{and } (a_j = b_j) \text{ for all } j \neq max(ki_a); \\ \delta, & \text{if } (a_i = b_i) \text{ for all } 1 \leq i \leq n \text{ and } (ki_a \neq ki_b) \\ & \text{and } ki_b \text{ is encoding of the } k \text{ paths with} \\ & \text{the largest bandwidths in network} \\ & \text{state } (a_1, a_2, \dots, a_n); \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

There are three types of transitions, one for flow departures, one for flow arrivals, and one for link state updates. The first type of transitions corresponds to the case when one current flow on path i leaves the system, $1 \leq i \leq n$. In this case, the transition rate from state $(ki_a, (a_1, \dots, a_{i-1}, a_i, a_{i+1}, \dots, a_n))$ to state $(ki_a, (a_1, \dots, a_{i-1}, a_i + 1, a_{i+1}, \dots, a_n))$ is $(C_i - a_i) \cdot \mu$. The second type corresponds to the case when a new flow arrives and is routed on path $max(ki_a)$. In this case, the transition rate from state $(ki_a, (a_1, \dots, a_{max(ki_a)} - 1, a_{max(ki_a)}, a_{max(ki_a)+1}, \dots, a_n))$ to $(ki_a, (a_1, \dots, a_{max(ki_a)} - 1, a_{max(ki_a)} - 1, a_{max(ki_a)+1}, \dots, a_n))$ is the connection arrival rate λ . The third transition corresponds to a link state update, where k new paths with the largest

bandwidths are computed. Let ki_b to be encoding of the k paths with the largest bandwidths in state $(ki_a, (a_1, \dots, a_n))$. The transition rate from state $(ki_a, (a_1, \dots, a_n))$ to state $(ki_b, (a_1, \dots, a_n))$ is δ . Using transition formula (4), it is straightforward to derive the infinitesimal generator for this Markov process and the equilibrium probabilities $\pi = (\pi(\vec{s}), \vec{s} \in S)$ can be computed in the similar way as those in the wsr with precise state information. Among all the states in the state space, the states that can result in blocking are the ones that do not have bandwidth on each of the k designated paths. The set of states, BS , that can result in blocking is

$$BS = \{(ki, (a_1, \dots, a_n)) : (ki, (a_1, \dots, a_n)) \in S\}$$

where $a_{i_m} = 0$ for all i_m in the decoding of ki . The blocking probability is equal to

$$B = \sum_{(ki, (a_1, \dots, a_n)) \in BS} \pi(ki, (a_1, \dots, a_n)).$$

Randomized Routing

The idea of randomized routing is to compute a set of feasible paths and then randomly select one for a connection. We will assume that wsr is used repeatedly to compute k feasible paths, where k is a parameter for the algorithm. Similar to multi-path routing, every time the link state is updated, k paths with the largest bandwidths are selected as the potential candidates to be used until the next link state update. When a flow arrives, it is randomly (with the uniform probability distribution) routed among the k paths. Since randomized routing does not probe the paths, it is possible that a request is routed to a path that does not have sufficient bandwidths, which results in a blocking. The Markov model for randomized routing is similar to that for multi-path routing: The system behaves differently only when a new request arrives. The state of the system is described by the vector

$$\vec{s} = (kindex, (r_1, \dots, r_n)).$$

Like in the model for multi-path routing, in addition to the current network state (r_1, \dots, r_n) , the k selected paths are recorded in $kindex$ in the system state. $kindex$ is some kind of encoding that records the k selected paths. By decoding $kindex$, k indices, i_1, i_2, \dots, i_k , for the k selected paths, can be derived. The state space is

$$S = \{(kindex, (r_1, r_2, \dots, r_n))\}$$

where $1 \leq kindex \leq \binom{n}{k}$ and $0 \leq r_1 \leq C_1, \dots, 0 \leq r_n \leq C_n$.

The number of states in S is equal to $\binom{n}{k} \prod_{l=1}^n (C_l + 1)$. Consider the transition from state $(ki_a, (a_1, a_2, \dots, a_n))$ to state $(ki_b, (b_1, b_2, \dots, b_n))$. Let decoding ki_a yield indices i_1, i_2, \dots, i_k . The state transition rate from state $(ki_a, (a_1, a_2, \dots, a_n))$ to state $(ki_b, (b_1, b_2, \dots, b_n))$ is expressed as (5).

The state transition is the same as that for multi-path routing except that when a new flow arrives, there is an equal probability for the flow to be routed to each of the k paths, which results in the $\frac{\lambda}{k}$ rate to the k corresponding states. Among all the states in

$$q((ki_a, (a_1, a_2, \dots, a_n)), (ki_b, (b_1, b_2, \dots, b_n))) = \begin{cases} (C_i - a_i) \times \mu, & \text{if } (ki_a = ki_b) \text{ and } (b_i = a_i + 1) \\ & \text{and } (a_j = b_j) \text{ for all } j \neq i; \\ \frac{\lambda}{k}, & \text{if } (ki_a = ki_b) \text{ and } (b_{i_m} = a_{i_m} - 1), \\ & \text{where } i_m \text{ is in the decoding of } ki_a \\ & \text{and } (a_j = b_j) \text{ for all } j \neq i_m; \\ \delta, & \text{if } (a_i = b_i) \text{ for all } 1 \leq i \leq n \text{ and } (ki_a \neq ki_b) \\ & \text{and } ki_b \text{ is encoding of the } k \text{ paths} \\ & \text{with the largest bandwidth in} \\ & \text{state } (a_1, a_2, \dots, a_n); \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

the state space, the states that can potentially result in blocking are the ones that do not have bandwidth on one of the k designated paths. The set of states, BS , that may potentially result in blocking is $BS = \{(ki, (a_1, \dots, a_n)) : (ki, (a_1, \dots, a_n)) \in S \text{ and there exists an } i_m \text{ in the decoding of } ki \text{ such that } a_{i_m} = 0\}$. The blocking probability is equal to

$$B = \sum_{(ki, (a_1, \dots, a_n)) \in BS} \pi(ki, (a_1, \dots, a_n)) \cdot \frac{zc}{k}.$$

Here, zc is the number of i_m 's in the decoding of ki such that $a_{i_m} = 0$. When the system is in such a state, a flow will be blocked when it is routed to a path that does not have sufficient bandwidths. Thus, the probability that a flow will be blocked is $\frac{zc}{k}$ and the overall blocking probability is

$$\sum_{(ki, (a_1, \dots, a_n)) \in BS} \pi(ki, (a_1, \dots, a_n)) \cdot \frac{zc}{k}.$$

A. Numerical Results

This sub-section studies the QoS routing algorithms using the analytical models. The findings are verified by the simulation in the next sub-section. To simplify the discussion, we will only present results for the fish topology with all paths having the same capacity. The topology can be represented with two parameters; the number of paths n and the capacity on each path Φ .

We will first consider the impacts of the imprecise network state information and examine the lower bound and the upper bound of the blocking probability of wsr . The lower bound is $E(\frac{\lambda}{\mu}, n\Phi)$ when the link state is precise and the upper bound is $E(\frac{\lambda}{\mu}, \Phi)$ when the link state is never updated and only one link is picked. We also show the blocking probability for the probability based methods, which is given by $E(\frac{\lambda}{\mu} \frac{1}{n}, \Phi)$. Fig. 3 shows the lower-bound and upper-bound for wsr , as well as the performance of the probability based routing. In this study, we fix the total capacity in the network, $n\Phi = 12$, and vary the number of paths (n). The load is fixed at $load = \frac{\lambda}{\mu} / (n\Phi) = 0.5$. There are several interesting observations in this figure. First, the difference between the upper bound and the lower bound is very large, which demonstrates that the impacts of imprecise network state information can be very significant. Second, the performance of probability based routing is much better than the

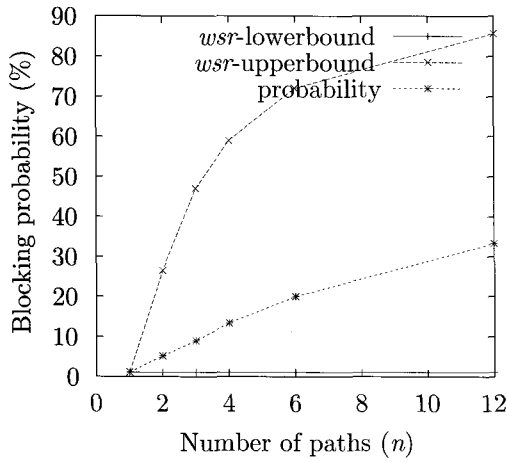


Fig. 3. Impacts of the number of paths ($n\Phi = 12$).

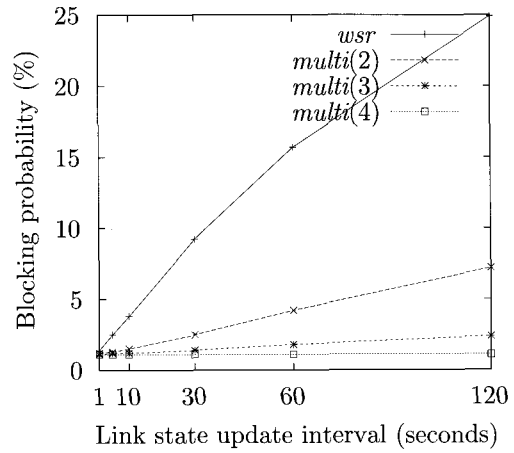


Fig. 5. Multi-path routing with different link state update intervals.

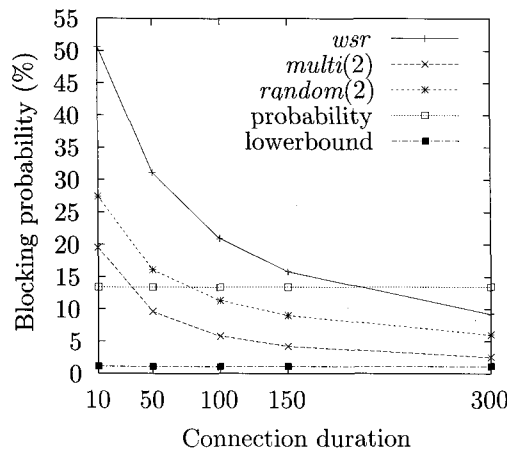


Fig. 4. Impacts of traffic patterns with imprecise state information.

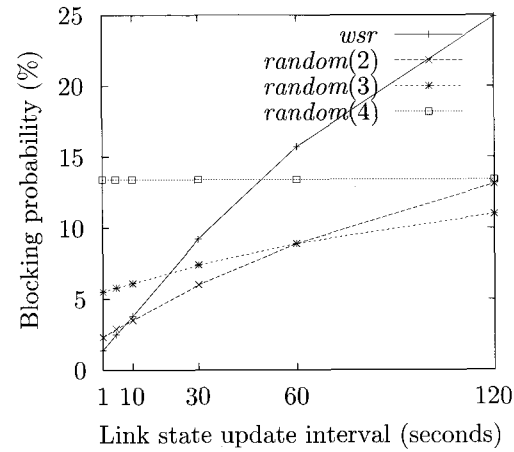


Fig. 6. Randomized routing with different link state update intervals.

upper-bound and much worse than the lower bound. The difference between the performance of probability based routing and the lower bound is essentially the difference between $E(\frac{\lambda}{\mu} \frac{1}{n}, \Phi)$ and $E(\frac{\lambda}{\mu}, n\Phi)$. This indicates that probability based routing is inherently limited in comparison to state based routing. Third, the number of paths can have significant impacts on the upper bound for *wsr* and on probability based routing. This indicates that the imprecise network state information will have more impacts on networks with higher connectivity (so that there are more paths between two nodes).

Fig. 4 shows the impacts of traffic patterns on the effectiveness of different routing algorithms when they handle the imprecise state information. In this study, $n = 4$ and $\Phi = 3$. The load is fixed at $load = \frac{\lambda}{\mu} / (n\Phi) = 0.5$ and the link state update interval $\frac{1}{\delta} = 30$ seconds. The connection holding time varies and the connection arrival rate varies accordingly. In the figure, we use legend *multi*(x) to denote multi-path routing that probes x paths simultaneously, and *random*(x) to denote randomized routing with x candidate paths. As can be seen from the figure, when the connection duration is short, the blocking probability is much higher for all availability based routing algorithms, including *wsr*, randomized routing, and multi-path routing. This figure also shows the advantage of probability based routing:

The performance of probability based routing is less sensitive to the traffic pattern. Thus, probability based routing offers an attractive alternative in the case when precise network state information is difficult to maintain, for example, when there are many short flows in the system.

Fig. 5 shows the performance of multi-path routing with different link state update intervals. In this study, $n = 4$, $\Phi = 3$, $\lambda = 0.02$, and $\frac{1}{\mu} = 300$ seconds. $load = \frac{\lambda}{\mu} / (n\Phi) = 0.5$. As can be seen from the figure, multi-path routing consistently out-performs *wsr* and the improvement is quite significant when the link state update interval is large. The multi-path routing algorithm achieves better performance when probing more paths. Note that *multi*(4) achieves the lower bound since $n = 4$. A similar trend is observed for other traffic patterns.

Fig. 6 shows the performance of randomized routing with different link state update intervals. In this study, $n = 4$, $\Phi = 3$, $\lambda = 0.02$, and $\frac{1}{\mu} = 300$ seconds. Thus, $load = \frac{\lambda}{\mu} / (n\Phi) = 0.5$. There are a number of interesting observations in this figure. First, the randomized routing does not always out-perform *wsr*. When $\frac{1}{\delta} = 5$ seconds, *wsr* achieves the best performance among all schemes in this study. Randomized routing achieves better performance when the link state update interval is large. Second, unlike multi-path routing, more candidate paths do not

necessarily improve routing performance. This indicates that finding the optimal number of candidate paths can be difficult. A similar trend is observed for other traffic patterns.

In summary, the following conclusions can be drawn from this numerical study. First, the imprecise network state information can have a very large impact on the performance of the traditional QoS routing scheme, *wsr*. The imprecise state information has more impacts when there are more paths between two nodes. Second, QoS routing algorithms that compute feasible paths purely based on the probability inherently have a limit on their performance. To effectively explore network resources, availability based routing schemes should be used. Third, multi-path routing significantly and consistently improves routing performance over its single-path routing counterpart. Finally, the performance of randomized routing is hard to predict. Sometimes, it may perform worse than *wsr*, the plain QoS routing algorithm.

B. Simulation Results

To verify the findings obtained with the analytical models and to evaluate the performance of the routing algorithms in practical situations on general topologies, we perform extensive simulations using a number of "Internet-like" random topologies with an average nodal degrees between 3 and 4. The "Internet-like" random topologies are generated using the *Georgia Tech Internetwork Topology Models (GT-ITM)* [17] with *edgmethod* = 3 (Edgmethod is a parameter in GT-ITM). The trend in the results for these random topologies is similar to that in the results for an ISP topology shown in Fig. 7. We will only present the results for the topology in Fig. 7 in this sub-section. 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 for the topology in Fig. 7 are $C = 2000$, $N = 18$, $L = 60$, and $h' = 2.36$. The link state information is updated periodically with an interval of $\frac{1}{\delta}$ seconds. Unless specified otherwise, the bandwidth requirement of a flow follows an exponential distribution with a mean value of $B = 300$. The average flow arrival rate, λ , is set depending upon the desired load.

The performance of widest-shortest routing, randomized routing, multi-path routing, and probability based routing schemes is studied. For randomized routing, we use the per pair path selection heuristic [18] to determine the set of feasible paths. The number of feasible paths to be selected in the randomized routing algorithm is 5. Multi-path routing also uses the per pair path selection heuristic to compute a number of feasible paths to be probed simultaneously. The maximum number of paths to be probed simultaneously by the multi-path routing algorithm is 5. For probability based routing, we assume that the

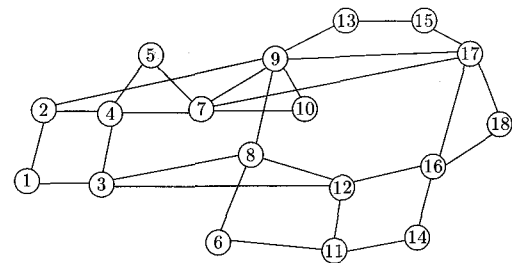


Fig. 7. The ISP topology.

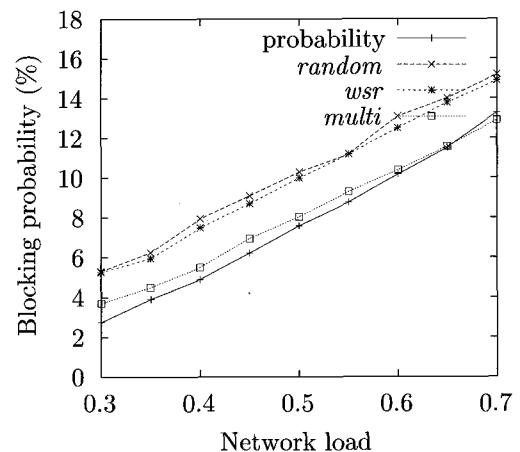


Fig. 8. Performance when the link state update interval is 5 seconds.

empirical availability independent probability distribution information is maintained using the approach in [8]. With the probability distribution, the shortest-safest routing algorithm [19] is used to compute feasible paths. A blocked flow is dropped without being retried. All the results are obtained with at least a 95% confidence level.

Fig. 8 shows the results when the link state update interval is small ($\frac{1}{\delta} = 5$ seconds). In this case, the network state information is fairly precise. As shown in the figure, for all different network loads, multi-path routing performs the best, followed by *wsr*, followed by the randomized routing algorithm. The probability based routing algorithm performs the worst. These results confirm the conclusions in the analytical study. First, when the network state information is sufficiently accurate, routing based resource availability out-performs routing based on resource availability probability. Second, randomized routing performs worse than the plain *wsr* when the state information is sufficiently precise. Third, multi-path routing consistently out-performs *wsr*. The improvement in this case is not very significant since *wsr* performs fairly well when the state information is precise.

Fig. 9 shows the results when the link state update interval is large ($\frac{1}{\delta} = 120$ seconds). In this case, the network state information is imprecise. Note that the average connection duration is 60 seconds. In this experiment, probability based routing results in the best performance. Multi-path routing performs significantly better than *wsr*. One result that is not observed in the analytical study is that randomized routing performs slightly worse than *wsr* even when the link state update interval is large.

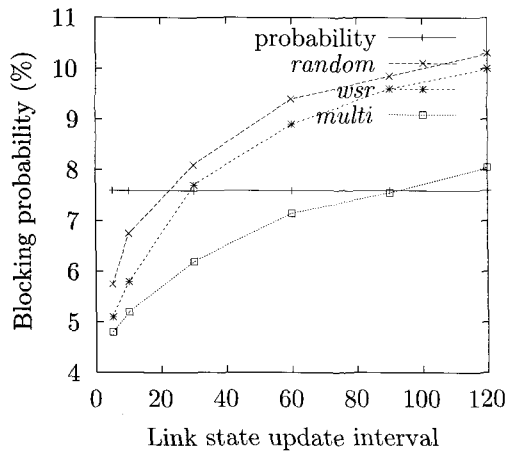


Fig. 9. Performance when the link state update interval is 120 seconds.

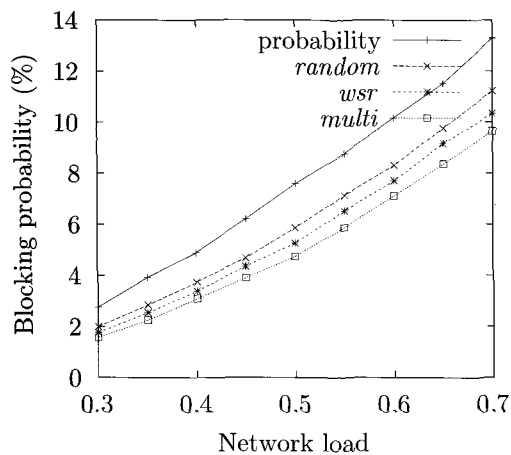


Fig. 10. Impacts of the link state update interval.

We have simulated other traffic patterns. The results are consistent. We did not observe in any case that the randomized routing algorithm *significantly* out-performs *wsr*. In most cases, randomized routing performs slightly worse than *wsr*. We believe that the reason for this is that in randomized routing some of the candidate paths are longer than the single path found by *wsr*. Thus, the side effect of randomized routing is that longer paths are sometimes selected for a connection request, which degrades the performance.

Fig. 10 shows the impacts of the link state update interval. The probability based scheme does not use the link state information to compute feasible paths and is not affected by the link state update interval. For all the availability based schemes, the link state update interval has a large impact. As the interval increases, the blocking probability for all the availability based schemes increases. As can be seen from the figure, multi-path routing tolerates the imprecise state information fairly well. Other experiments, for different network topologies, different traffic patterns, and different network loads, have been carried out. The results show a similar trend.

In the simulation study, we observe the findings obtained from the analytical models:

1. The imprecise network state information can have a very

large impact on the performance of the traditional QoS routing scheme, *wsr* (from the relative performance of *wsr* in Figs. 8 and 9).

2. Probability based routing performs well only when the network state information is very imprecise (Figs. 8 and 9).
3. Multi-path routing significantly and consistently improves routing performance over its single-path routing counterpart (Figs. 8 and 9).
4. Randomized routing may not be effective (Figs. 8, 9, and 10).

Although our analytical models are derived with many simplified assumptions such as the fish topology, the Poisson arrival rate, the same bandwidth requirement for all requests, etc, they give valuable insights about the routing algorithms in practical situations when many of the assumptions are relaxed. This demonstrates the power of the models.

IV. LOCAL REROUTING

As can be seen from the previous section, probing multiple paths is a robust mechanism to handle imprecise network state information. Explicitly probing multiple paths, however, can either introduce much larger overheads in comparison to single path routing when multiple paths are probed concurrently or result in a long path setup time when multiple paths are probed sequentially. We propose a new mechanism, called Local Rerouting (LR), that allows multiple paths to be probed without the limitations of the traditional multi-path routing schemes. LR can be incorporated in the resource reservation process to improve the effectiveness of path establishment in the presence of imprecise network state information. The idea is to apply the concept of dynamic rerouting during the path establishment process. Dynamic rerouting is a fault-tolerance mechanism that deals with link failures in operating networks. In this section, we show that such a concept can be extended to improve the effectiveness of QoS path establishment.

LR assumes that each router can compute by-pass paths for each link directly connected to the router. Different methods to compute the by-pass paths will be discussed later in the section. Once the by-pass paths are computed, local rerouting can apply at the resource reservation process: When a router tries to reserve bandwidth on a link for a connection, if the bandwidth on the link cannot satisfy the QoS requirement (this is due to the fact that the QoS routing algorithm uses stale link state information to compute the path), instead of failing the connection request as in the traditional resource reservation scheme, local rerouting will examine the by-pass paths to determine whether there is a by-pass path that can satisfy the QoS requirement. When such a by-pass path exists, resources will be reserved along the by-pass path and the reservation process will continue.

There are two key issues in local rerouting: Computing the by-pass paths and deciding which by-pass path can be used to replace the link with insufficient bandwidth. Since by-pass paths are used to protect a link, they are "local" to the router connecting the link in that all the links in the by-pass paths are a few hops away from the router. Thus, computing and deciding a by-pass path can be done with some "local" mechanism that is

scalable to the network size. The following three schemes can be used to support local rerouting:

- **Local link state update:** This scheme augments the link state update mechanism. Using this scheme, there are two levels of link state updates; the global (network wide) link state updates that follow the traditional link state algorithm (e.g., the link state information is updated periodically) and local link state updates where local bounded flooding is used. For example, in local link state updates, a router may distribute its link states to routers within X hops. Since local link state updates are distributed locally, it is possible to use it to maintain precise “local” information, that is, each router may have precise state information for links that are within X hops to the router. A by-pass path that can satisfy the QoS requirement can be computed by a router when needed using the precise “local” information. When the local flooding is bounded to X hops, the maximum length of a by-pass path is $X + 1$.
- **On-demand local link state update:** This scheme does not require the link state maintenance mechanism to be changed. At the time when a router decides that a link does not have sufficient bandwidth, it sends out requests for the neighboring routers asking for precise link state information. After the neighbors reply, the router can then compute the by-pass paths. On-demand local link state update is the reactive counterpart of the local link state update scheme.
- **Pre-computed by-pass paths:** In this scheme, a set of pre-computed by-pass paths for each link is maintained at each router. At the time when a router decides that a link does not have sufficient bandwidth, it probes the set of by-pass paths sequentially to decide if there exists a by-pass path that has sufficient bandwidth.

The maximum length of by-pass paths, ML , is an important parameter for local rerouting. It not only decides the effectiveness of local rerouting, but also determines the overheads of the scheme. Consider the case when $ML = 2$ (at most 2 links in a by-pass path). In this case, the overhead is small: Only one hop bounded flooding is needed to support the local link state update scheme. However, depending on the network topology, the number of links that are protected by by-pass paths may be limited and local rerouting may not be effective. Hence, to decide the best ML for a network, a compromise between link coverage and overheads for computing by-pass paths must be determined.

Local rerouting allows multiple paths between the source and destination to be probed, which increases the possibility of a successful path establishment in the presence of imprecise state information. In practice, most links along the path computed by a QoS routing algorithm are likely to satisfy the QoS requirement. In the rare cases when some links in the path cannot satisfy the QoS requirement, local rerouting can fix the problem inexpensively. Compared to the scheme that concurrently probes multiple paths, local rerouting introduces less signaling overheads. Compared to the scheme that probes multiple paths sequentially, local rerouting has a shorter path setup time. In addition, since the by-pass paths are computed based on the precise (local) network state information, the effectiveness of LR is

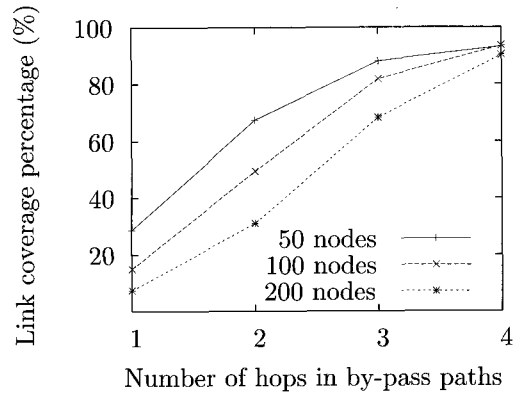


Fig. 11. Link coverage percentage

not sensitive to imprecise global network state information.

There are some potential issues with local rerouting. For example, LR replaces a link with a longer by-pass path. This can potentially increase the resources used by a connection and degrade the overall network performance. LR also introduces overheads in the by-pass path computation. These issues are investigated in the performance evaluation.

A. Performance Study

To study the effectiveness of local rerouting (LR) in handling imprecise network state information, we evaluate the performance when LR is applied to widest-shortest routing (*wsr*), randomized routing, multi-path routing, and probability based routing. Note that one advantage of LR is that it can be combined with almost all existing schemes that were designed to deal with imprecise state information. In the experiments, the path computation methods for different routing schemes are the same as those described in Section III-B.

We perform extensive simulations using “Internet-like” random topologies with an average nodal degrees between 3 and 4 (generated by GT-ITM) as well as the ISP topology in Fig. 7. The simulation settings are the same as those described in Section III-B. The trend in different Internet-like topologies is similar. Hence, unless specified otherwise, we will present the results for a 100-node random topology with the following parameters: $N = 100$, $L = 428$, $h' = 3.30$, and $C = 4000$. The network traffic used in the simulation has the following parameters; the mean bandwidth requirement $B = 200$ and connection duration $\frac{1}{\mu} = 60$ seconds.

In order for LR to be effective, links in the network must be protected by local by-pass paths. Fig. 11 shows the percentage of links in a network that are protected by at least one by-pass path. We show the results for networks with 50, 100, and 200 routers. The results are the average of 50 random networks generated by GT-ITM with the same parameter (an average nodal degree of 4). As can be seen from the figure, having the by-pass paths of length 3 or 4 can provide sufficient coverage for the links in the networks. For example, with the maximum by-pass path length of 4, more than 90% of links are protected in all cases. In the rest of the section, the maximum length of by-pass paths is set to be 4.

Fig. 12 shows the results when LR is applied to *wsr* on the

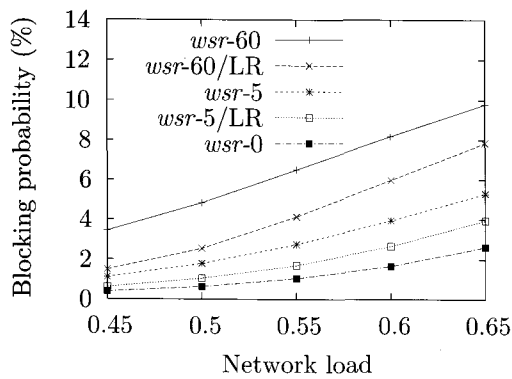


Fig. 12. Results for combining local rerouting with wsr.

100-node random network when the link state update intervals ($\frac{1}{\delta}$) are 5 seconds and 60 seconds. In the figure, the legend $wsr - X$ represents wsr with a link state update interval of X seconds. The legend $wsr - X/LR$ represents $wsr - X$ with LR. The rest of this section follows the same convention. The $wsr - 0$ scheme represents the case when precise network state information is used. As can be seen from the figure, in both cases, local rerouting is effective in improving the blocking probability, which indicates that LR is effective for different link state update intervals. The performance of $wsr - 5/LR$ is quite close to that of $wsr - 0$.

Fig. 13 shows the improvement percentage on blocking probability when LR is used with different routing algorithms. Let b_s be the blocking probability for routing scheme s , the improvement percentage is computed by $\frac{b_s - b_{s/LR}}{b_{s/LR}} \times 100$. This experiment assumes a 30 seconds link state update interval. Routing algorithms including wsr , multi-path routing, probability based routing, and randomized routing are considered. As can be seen from the figure, large improvements are obtained for wsr , probability based routing, and randomized routing (17% to 120% depending on the network load). For multi-path routing, the improvement is relatively small, but still significant (2% to 25%). This is mainly due to the fact that both approaches explore multiple paths for establishing a connection. Thus, there are some overlaps among the paths explored by both approaches, which renders the combination of LR and multi-path routing less effective than the combination of LR and other schemes. This figure also shows that local rerouting is more effective when the network load is low. There are two reasons for this. First, when the network load is low, there are more rerouting opportunities and local rerouting can more effectively handle ineffective routing decisions. Second, local rerouting uses a longer path for a connection, which usually results in the decreasing of effectiveness when the network load becomes higher.

The results in Figs. 12 and 13 show that local rerouting is very effective in reducing the blocking probability for QoS path establishment. Another aspect in QoS path establishment is the overhead introduced by the QoS routing and path establishment mechanisms. LR can be supported with three schemes: Local link state update, on-demand link state update, and pre-computed by-pass paths. With local link state updates and on-demand link state updates, link state maintenance overheads

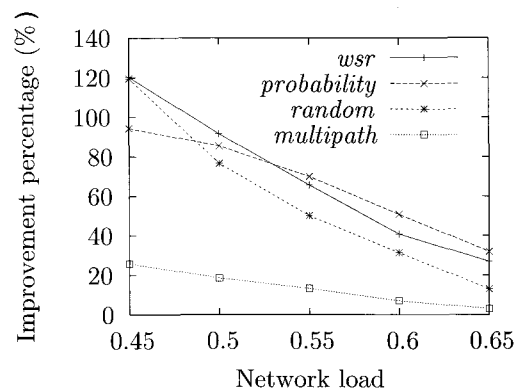


Fig. 13. Improvement of local rerouting on blocking probability.

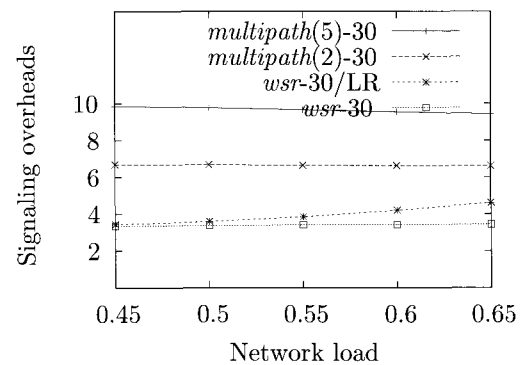


Fig. 14. Signaling overheads for the pre-computed by-pass paths scheme (average number of links probed per established connection).

are introduced. When one of these schemes is used, the signaling overhead introduced by LR is only in forms of longer paths. With the pre-computed by-pass paths scheme, the link state maintenance overhead remains the same as the traditional scheme. However, since a router may probe multiple by-pass paths to find the one that can satisfy the QoS constraints, the signaling overhead may be significant.

Fig. 14 compares the signaling overhead of the pre-computed by-pass paths scheme with that of other routing schemes. The signaling overhead is defined as the average number of links that must be probed to establish a connection. Using pre-computed by-pass paths, a router may need to probe multiple by-path paths to find one that satisfies the QoS requirement. The figure shows the overheads for $wsr - 30$, $wsr - 30/LR$, $multipath(2) - 30$ (multi-path routing with 2 paths at most for each connection), and $multipath(5) - 30$ (multi-path routing with 5 paths at most for each connection). As can be seen from the figure, local rerouting introduces slightly more overheads than single path routing while having much lower overheads than multi-path routing, even 2-path routing. This is because most of the links in the computed paths have sufficient resources and no rerouting is needed. These results indicate that LR with pre-computed by-pass paths only introduces minor overheads in comparison to multi-path routing.

Let us consider the link state maintenance overheads when local link state update or on-demand local link state update is used. Different ways to distribute link state update affect the

Table 1. Link state maintenance overheads.

load	wsr-30	wsr-30/LR		wsr-12
		local update	on-demand	
0.45	3.3	89.0	3.8	8.3
0.50	3.3	100.0	4.3	8.3
0.55	3.3	110.7	5.2	8.3
0.60	3.3	121.7	6.4	8.3
0.65	3.3	132.0	7.7	8.3

Table 2. Blocking probability.

load	wsr-30	wsr-30/LR	wsr-12	wsr-0
0.45	2.49%	1.13%	1.62%	0.39%
0.50	3.64%	1.90%	2.55%	0.64%
0.55	5.07%	3.06%	3.73%	1.05%
0.60	6.59%	4.68%	5.08%	1.66%
0.65	8.21%	6.47%	6.52%	2.62%

maintenance overheads. We use an abstract overhead measurement without considering the details about how the link state information is distributed. For different schemes, the overhead is defined as the number of updates each node receives per second. For example, for the 100-node network, using the timer based scheme with a link state update interval of 30 seconds, each node must receive 99 updates in 30 seconds and the overhead is thus $99/30 = 3.3$. When LR is employed, the overheads will depend on other factors such as the network load. Table 1 shows the link state maintenance overheads and Table 2 shows the corresponding performance (blocking probability) for $wsr - 30$, $wsr - 30/LR$, and $wsr - 12$. As can be seen from the table, the pro-actively distributing local link state information can introduce large overheads: 89.0 updates per second versus 3.3 updates per second without LR at the network load of 0.45. To use this scheme, some mechanisms must be employed to further reduce the overheads. One possibility is to use the threshold based local updates. On the other hand, on-demand local link state updates introduce much less overheads compared to the pro-active updates. This is because the on-demand updates are invoked much less frequently than proactive updates. Overall, $wsr - 30/LR$ achieves lower blocking probability, but has higher overheads than $wsr - 30$. Since path establishment in the presence of imprecise state information is essentially a compromise between performance and overheads, we compare $wsr - 30/LR$ with $wsr - 12$. With on-demand local link state updates, $wsr - 30/LR$ introduces less overheads in maintaining the link state information while achieving lower blocking probability for all the network loads studied. This indicates that incorporating LR is more effective than simply decreasing the link state update interval to improve performance.

Among the three schemes proposed for computing local by-pass paths; pro-actively distributing local link state updates introduces large overheads. However, LR with on-demand link state updates or the pre-computed by-pass paths scheme is effective; the blocking probability is significantly improved without introducing excessive overheads.

V. CONCLUSION

In this paper, we study the problem of establishing QoS paths in the presence of imprecise global network state information. We develop analytical models for QoS routing algorithms and analyze various QoS routing schemes that were designed to tolerate imprecise network state information. We propose and evaluate a new scheme, local rerouting, that allows multiple paths to be probed for establishing a connection. The results of our performance study show that local rerouting is effective in improving path establishment performance while only introducing minor overheads.

REFERENCES

- [1] L. Zhang, S. Deering, D. Estrin, S. Shenker and D. Zappala, "RSVP: A new resource reservation protocol", *IEEE Network*, vol. 31, no. 9, pp. 8–18, Sept. 1993.
- [2] R. Guerin and A. Orda, "QoS-based routing in networks with inaccurate information: Theory and algorithms.", in *Proc. IEEE INFOCOM'97*, Apr. 1997, pp. 75–83.
- [3] 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.
- [4] G. Apostolopoulos, R. Guerin, S. Kamat, and S. Tripathi, "Improving QoS routing performance under inaccurate link state information," in *Proc. the 16th Int'l Teletraffic Congress*, June 1999, pp. 1351–1362.
- [5] D. Ghosh and R. Acharya, "A probabilistic approach to hierarchical QoS routing," in *Proc. IEEE Int'l Conf. Computer Networks*, Oct. 2001, pp. 416–421.
- [6] T. Korkmaz and M. Krunz, "Bandwidth-delay constrained path selection under inaccurate state information," *IEEE/ACM Trans. Networking*, vol. 11, no. 3, pp. 384–398, June 2003.
- [7] D. H. Lorenz and A. Orda, "QoS routing in networks with uncertain parameters," *IEEE/ACM Trans. Networking*, vol. 6, no. 6, pp. 768–778, Dec. 1998.
- [8] X. Yuan and G. Yang, "Empirical probability based QoS routing," in *Proc. IEEE ICC*, May 2003, pp. 1713–1717.
- [9] Y. Zheng, T. Korkmaz, and W. Dou, "Two additive-constrained path selection in the presence of inaccurate state information," in *Proc. the 1st EuroNGI Conf. Next Generation Internet Networks (NGI 2005)*, Apr. 2005, pp. 1–8.
- [10] S. Chen and K. Nahrstedt, "Distributed QoS routing with imprecise state information," in *Proc. IEEE Int'l Conf. Computer and Commun. Networks*, Oct. 1998, pp. 614–621.
- [11] Y. Jia, I. Nikolaidis, and P. Gburzynsku, "Multiple path routing in networks with inaccurate link state information," in *IEEE ICC*, June 2001, pp. 2583–2587.
- [12] Y. Zhong and X. Yuan, "Impact of resource reservation on the distributed multi-path Quality of Service routing scheme," in *Proc. Int'l Workshop on Quality of Service (IWQoS)*, June 2000, pp. 95–104.
- [13] J. Wang, W. Wang, J. Chen, and S. Chen, "A randomized QoS routing algorithm on networks with inaccurate link-state information," in *Proc. IEEE Int'l Conf. Commun. Technol.*, 2000, pp. 1617–1622.
- [14] ATM Forum, Private network network interface (PNNI) v1.0 specifications, June 1996.
- [15] X. Yuan, W. Zheng and S. Ding, "A comparative study of QoS routing algorithms that tolerate imprecise state information," in *Proc. IEEE Int'l Conf. Computer and Commun. Networks*, Oct. 2002, pp. 230–235.
- [16] Q. Ma and P. Steenkiste, "Quality-of-Service routing for traffic with performance guarantees," in *Proc. 4th Int'l IFIP Workshop on Quality of Service*, May 1997, pp. 115–126.
- [17] E. W. Zegura, K. Calvert, and S. Bhattacharjee, "How to model an inter-network," in *Proc. IEEE INFOCOM*, Mar. 1996, pp. 594–602.
- [18] X. Yuan and A. Saifee, "Path selection methods for localized Quality of Service routing," in *Proc. IEEE Int'l Conf. Computer and Commun. Networks*, Oct. 2001, pp. 102–107.
- [19] G. Apostolopoulos, R. Guerin, S. Kamat, and S. Tripathi "Quality of Service based routing: A performance perspective", *ACM SIGCOMM Computer Commun. Rev.*, vol. 28, no. 4, pp. 17–28, 1998.



Xin Yuan received the B.S. and M.S. degrees in Computer Science from Shanghai Jiaotong University in 1989 and 1992, respectively. He received the Ph.D. degree in Computer Science from the University of Pittsburgh in 1998. He is currently an associate professor at the Department of Computer Science, Florida State University. His research interests include parallel and distributed systems, cluster computing, networking, and system/network quality of service. He is a member of IEEE and ACM.



Zhenhai Duan received the B.S. degree from Shandong University, China, in 1994, the M.S. degree from Beijing University, China, in 1997, and the Ph.D. degree from the University of Minnesota, in 2003, all in Computer Science. He is currently an assistant professor in the Computer Science Department at the Florida State University. His research interests include computer networks and multimedia communications, especially Internet routing protocols and service architectures, scalable network resource control and management, and network security. He is a co-recipient of the 2002 IEEE International Conference on Network Protocols (ICNP) Best Paper Award and the 2006 IEEE International Conference on Computer Communications and Networks (ICCCN) Best Paper Award. He is a member of IEEE and ACM.