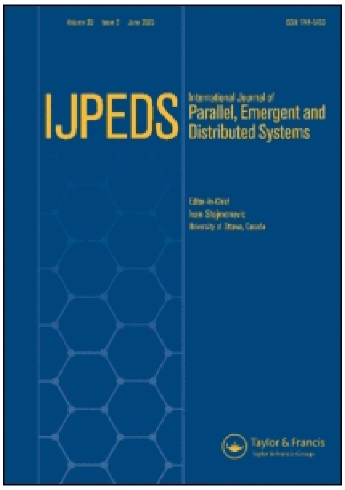


International Journal of Parallel, Emergent and Distributed Systems

Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713729127

SMRS: A scalable multi-path routing scheme

Quan Yuan [a]; Jie Wu [a]; Ionut Cardei [a]
[a] Department of Computer Science and Engineering, Florida Atlantic University, Boca Raton, FL, USA

Online Publication Date: 01 February 2009

To cite this Article Yuan, Quan, Wu, Jie and Cardei, Ionut(2009)'SMRS: A scalable multi-path routing scheme',International Journal of Parallel, Emergent and Distributed Systems,24:1,69 — 84

To link to this Article: DOI: 10.1080/17445760802310975

URL: http://dx.doi.org/10.1080/17445760802310975

PLEASE SCROLL DOWN FOR ARTICLE

International Journal of Parallel, Emergent and Distributed Systems
Vol. 24, No. 1, February 2009, 69–84

SMRS: A scalable multi-path routing scheme

Quan Yuan*, Jie Wu[1] and Ionut Cardei

Department of Computer Science and Engineering, Florida Atlantic University, Boca Raton, FL 33431, USA

(*Received 11 December 2007; final version received 12 June 2008*)

Source routing has been extensively investigated as a method of improving diversity in path selection. Due to its limited scalability, the application of source routing is hindered today. Yang and Wetherall proposed a tag-based routing architecture to provide a set of non-shortest-path routes as an alternative to explicit source routes. The idea is that each router constructs a deflection set with the 1-hop route information and utilises tags as hints to independently deflect packets to the neighbours in the deflection set. To find more diverse paths, one approach is to increase the route knowledge of every node. However, there is a tradeoff in that more route knowledge leads to more diverse loop-free and dead end-free paths, but it also causes more overhead. In this paper, our objective lies in designing a method to find more diverse paths, with minimum knowledge. In addition, the tag-based deflection approach does not consider the length of the utilised paths as a main issue, which may lead to long paths. With this in mind, we propose a scalable multi-path routing scheme (SMRS), which consists of two components. One is the extension of the Yang and Wetherall approach to 2-hop history information. The other is the next hop selection algorithms to control the average length of the utilised paths. Comprehensive simulations are conducted to compare the performance of the tag-based deflection routing and SMRS.

Keywords: multi-path routing; routing diversity; routing scalability; path selection

1. Introduction

Because of the dynamic network environment, single-path routing, which usually constructs the shortest path, is subject to network faults, e.g. link or router failure, link congestion, low capacity link and high delay link, resulting in degraded delivery efficiency. Thus, finding alternative paths between the source and the destination, which is mainly implemented by multi-path routing protocols, becomes necessary. Compared to single-path routing, multi-path routing provides more flexibility and diversity in path selection. In addition, multi-path routing alleviates the influence of exceptions in networks by providing alternative paths for packet forwarding, which improves the reliability and efficiency of networks.

One of the main challenges for multi-path routing protocols is how to discover available loop-free paths between a pair of nodes. Recent research focuses on source routing, which argues that, given a destination, the source can find all the possible loop-free paths to it after locally mapping a global network with some tools, e.g. trace route. Then, the source determines several paths for packet delivery and stores the corresponding routing instructions in the forwarded packet. Once the packet is in the network, it will traverse strictly based on the instructions. However, this approach is unscalable because maintaining the global network map on each

*Corresponding author. Email: qyuan@fau.edu

ISSN 1744-5760 print/ISSN 1744-5779 online

DOI: 10.1080/17445760802310975
http://www.informaworld.com

router leads to large overhead, especially when the network size grows. Moreover, source routes tend to be relatively static and therefore do not fit the dynamic hop-by-hop routing facilities.

In [16], Yang and Wetherall presented a tag-based routing architecture, which uses routing deflections to provide path diversity without the scalability issue associated with the explicit source routes. Instead of finding all the possible loop-free paths, the proposed scheme provides a small set of diverse paths for packet delivery with one hop information – neighbours' shortest-path costs to a given destination. Specifically, they developed three deflection rules that enable routers to independently deflect packets and thereby collectively construct a set of diverse loop-free and deadend-free paths. The source can affect the choice of deflection at each router through setting tags in the forwarded packets. Because routing deflections build on the shortest path machinery and the source does not need to construct the global network map for discovering paths, their scheme exhibits good scalability.

To find more diverse paths between a node pair, one approach is to increase the route knowledge of every node in the network. However, there is a tradeoff in that more route knowledge leads to more diverse paths, but it also causes more overhead. We aim at designing a method to find more diverse loop-free and deadend-free paths, with minimum knowledge. Additionally, in the Yang and Wetherall approach, routers deflect packets to a randomly selected qualified neighbour in nature, even though the source can set a tag value in the packet to affect its deflection choice and use TTL to determine when to apply deflection. Unfortunately, there is the problem that the path used to deliver data may become too long.

In this paper, we propose a scalable multi-path routing scheme (SMRS) to address the above problems. In SMRS, every node employs two-hop history information and one hop forward information to construct a local successor set, after receiving a packet. Then, the node chooses a neighbour from the successor set as the next hop to forward the packet. This way, a set of diverse paths are discovered for routing. SMRS consists of two components. One is the extension of the Yang and Wetherall deflection rules with one more history hop route information to construct the local successor set at each node. The other is the next hop selection algorithms to control the average length of the utilised paths. We prove that, with only one more history hop route information, the performance (in terms of routing diversity) improves substantially. Also, we find that 2-hop history information based schemes outperform other *k*-hop history information based schemes. Our approach has the following features:

- *Diversity*: By the comprehensive simulations, we prove that SMRS can discover a larger set of diverse paths between a node pair in the networks than the 1-hop route information based schemes. Thus, routing diversity is further improved, and traffic congestion can avoided.
- *Scalability*: SMRS builds on shortest-path machinery without maintaining the global topology information. In addition, besides the knowledge of 1-hop routing condition, we utilise one more history hop route information to implement SMRS. These two properties guarantee that SMRS has nearly the same scalability as the Yang and Wetherall approach.
- *Low Latency:* When using link delay as path cost, packets tend to be delivered on short paths, even though SMRS can provide a sufficient number of diverse loop-free paths. Long paths are mainly used as backup when short paths do not work.

This paper is organised as follows. Section II presents two extended rules of the Yang and Wetherall deflection rules for the local successor set construction. Section III discusses the routing schemes based on the successor set. Section IV provides simulation results, and Section V discusses the related work. We conclude our work in Section VI.

2. Extended deflection rules

Our rules are built on the shortest-path costs. The notation we use to define rules is detailed in Table 1, where the computation of shortest path costs $D_i^j(G)$ between node i and node j can be computed as an extension of the routing protocol they run for base routing. For example, each node has the distance and next hop knowledge for any destination with the base routing table. A node could acquire its neighbours' shortest-path cost information by exchanging such knowledge with its neighbours. Thus, compared to the base routing algorithm, our solution only needs trivial extra cost.

2.1 Review of deflection rules

Yang and Wetherall define the following three deflection rules with one hop route information, which means node i keeps the route information of node $i - 1$ and node $i + 1$, e.g. shortest path costs to a given destination, as in Figure 1(a).

Rule 1: The deflection set for a node i is those neighbours $i + 1$ for which $D_{i+1}^d(G) < D_i^d(G)$.

Rule 2: The deflection set for a node i is those neighbours $i + 1$ for which either of the following conditions is satisfied: (1) $D_{i+1}^d(G) < D_i^d(G)$; (2) $D_{i+1}^d(G) < D_{i-1}^d(G)$.

Rule 3: The deflection set for a node i is comprised of the neighbours $i + 1$ for which node $i + 1$ is different from node $i - 1$ and either: (1) $D_{i+1}^d(G \backslash l_{i+1}) < D_i^d(G \backslash l_i)$; (2) $D_{i+1}^d(G \backslash l_{i+1}) < D_{i-1}^d(G)$.

Rule 1 regulates that the cost to a given destination at each router along a path decreases strictly and will eventually become zero. Rule 2 allows that the cost to a destination can go down in two hops. Rule 3 provides some complement to Rule 2 by finding an alternative to immediate backtracking. These three rules provide a set of loop-free (a safety condition) and deadend-free (a liveness condition) paths.

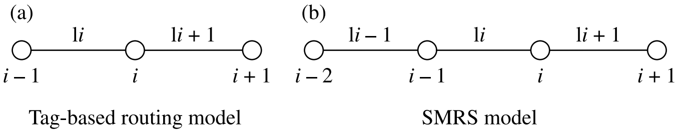
2.2 Challenges

Our challenges lie in designing a method to find more diverse loop-free and deadend-free paths, with minimum knowledge. A trivial extension of the deflection rules is to increase each node's route knowledge to two hops and revise the corresponding rules. However, such an approach costs too much overhead, e.g. memory. Suppose the average node degree is n. The overhead increases to $(n/2)^2$, compared to $n/2$ which is the overhead of the 1-hop information based scheme.

In SMRS, we extend the deflection rules with the route information of only one more history hop, as in Figure 1(b). We find that although the change is subtle, the performance improves substantially. Specifically, node i still keeps the route information of node $i - 1$ and node $i + 1$. In addition, we assume that packets can store the route information of one passing node.

Table 1. The notations.

Notation	*Meaning*
D	Destination node
i	Sequence of nodes along a path, $i > 0$
l_i	Link that connects node $i - 1$ and node i
G	Graph of network
$G \backslash l$	Graph of network with link l removed
$D_i^j(G)$	Shortest path cost from node i to node j in network G
S_i	Successor set of node i



Tag-based routing model

SMRS model

Figure 1. (a) Presents the model of tag-based routing, in which node i keeps the route information of node $i-1$ and node $i+1$. (b) Illustrates SMRS model, in which node i knows the route information of node $i-1$ and node $i+1$ plus the route information of node $i-2$ is piggybacked to node i.

Thus, the route information of node $i-2$ can be piggybacked by the forwarded packets to node i. Note that if node $i-2$ or node $i-1$ does not exist, $D^d_{i-2}(G)$ or $D^d_{i-1}(G)$ will be infinite. Considering that in the Yang and Wetherall deflection rules, Rule 1 serves as a strawman for assessing the strength of other rules and Rule 2 includes all the choices provided by Rule 1, we only extend Rule 2 and Rule 3 in our case. In this paper, Rule E1 and Rule E2 are proposed as the extensions of deflection Rule 2 and Rule 3, respectively.

The main issue with the two extended rules is to prove they are loop-free and deadend-free. Both of these properties assure that paths discovered can always reach a given destination.

2.3 Rule E1

In our first rule, we regulate that nodes in the successor set should satisfy that the shortest path cost to the destination at each node along a path decreases in three hops.

Rule E1: Given the destination node d, the successor set S_i at node i includes neighbours $i+1$ which satisfy either of the following conditions:

$$D^d_{i+1}(G) < D^d_i(G), \tag{1}$$

$$D^d_{i+1}(G) < D^d_{i-1}(G), \tag{2}$$

$$D^d_{i+1}(G) < D^d_{i-2}(G). \tag{3}$$

Theorem 1. All the paths discovered by Rule E1 between any pair of nodes are loop-free and deadend-free.

Proof. To prove its loop-free, we present that any two directional adjacent links in the path would be visited at most twice. First, we define the cost of two directional adjacent links m_i as $\text{cost}(m_i) = \max\{D^d_i(G), D^d_{i+1}(G), D^d_{i+2}(G)\}$, where m_i is the two links connecting node i, $i+1$ and $i+2$. $\text{cost}(m_i)$ is the maximum shortest path cost of those three nodes to the destination d. According to Rule 1

$$\begin{aligned}\text{cost}(m_{i+1}) &= \max\left\{D^d_{i+1}(G), D^d_{i+2}(G), D^d_{i+3}(G)\right\}\\ &\le \max\left\{D^d_i(G), D^d_{i+1}(G), D^d_{i+2}(G)\right\} = \text{cost}(m_i),\end{aligned}$$

$$\begin{aligned}\text{cost}(m_{i+2}) &= \max\left\{D^d_{i+2}(G), D^d_{i+3}(G), D^d_{i+4}(G)\right\}\\ &\le \max\left\{D^d_i(G), D^d_{i+1}(G), D^d_{i+2}(G)\right\} = \text{cost}(m_i),\end{aligned}$$

because

$$\begin{aligned}
D^d_{i+3}(G) &< \max\left\{D^d_i(G), D^d_{i+1}(G), D^d_{i+2}(G)\right\} \\
D^d_{i+4}(G) &< \max\left\{D^d_{i+1}(G), D^d_{i+2}(G), D^d_{i+3}(G)\right\} \quad \le \max\left\{D^d_i(G), D^d_{i+1}(G), D^d_{i+2}(G)\right\} \\
D^d_{i+5}(G) &< \max\left\{D^d_{i+2}(G), D^d_{i+3}(G), D^d_{i+4}(G)\right\} \quad \le \max\left\{D^d_i(G), D^d_{i+1}(G), D^d_{i+2}(G)\right\},
\end{aligned}$$

then

$$\begin{aligned}
\mathrm{cost}(m_{i+3}) &= \max\left\{D^d_{i+3}(G), D^d_{i+4}(G), D^d_{i+5}(G)\right\} \\
&< \max\left\{D^d_i(G), D^d_{i+1}(G), D^d_{i+2}(G)\right\} = \mathrm{cost}(m_i).
\end{aligned}$$

Thus, we have $\mathrm{cost}(m_{i+q}) \le \mathrm{cost}(m_i)$ (for $q \ge 1$). Specifically, when $q \ge 3$, the cost of any two directional adjacent links m_{i+q} is strictly less than m_i, which implies that when $q \ge 3$, no two directional adjacent links would ever be the same. Moreover, because links l_{i+1} and l_{i+2} begin with different nodes, m_i and m_{i+1} cannot be the same two directional adjacent links. For $q = 2$, if node i is different from node $i + 2$, m_i and m_{i+2} begins with different nodes. Therefore, they are the different two directional adjacent links. Otherwise, m_i and m_{i+2} might be the same two directional adjacent links. Thus, any two directional adjacent links in the path constructed by Rule E1 would be visited at most twice. Intuitively, if we consider each link as a 'node' and imagine there is a 'link' between two adjacent 'nodes', an analogous form of deflection Rule 2 can be derived. According to deflection Rule 2, if 'nodes' along the path are different from each other, no directional 'link' can be re-visited on the forwarding path and there are no 'link'-level loops in Rule E2. Two directional adjacent links may be revisited only when 'node' overlapping occurs.

To prove deadend-free, shortest-path neighbours are always valid for Rule E1, which guarantees that the successor set will never be empty at any node. This is because, according to the definition of shortest-path neighbours, they always satisfy the condition $D^d_{i+1}(G) < D^d_i(G)$. □

Note that Rule E1 allows node i to return the packet to node $i - 1$ in some cases, which should be avoided, because returning the packet extends the path length without improving the path diversity. In addition, avoiding immediate moving-back can prevent the case that some links are visited twice. Thus, node $i - 1$ is taken into the local successor set only when there are no other nodes in it.

An example of Rule E1 is shown in Figure 2, where node S is the source node and node D is the destination. Applying Rule E1, paths discovered between S and D include path (S, F, D), path (S, A, F, D), path (S, A, B, D), and path (S, A, B, C, E, D), while path (S, A, B, C, E, D) cannot be found by deflection Rule 2.

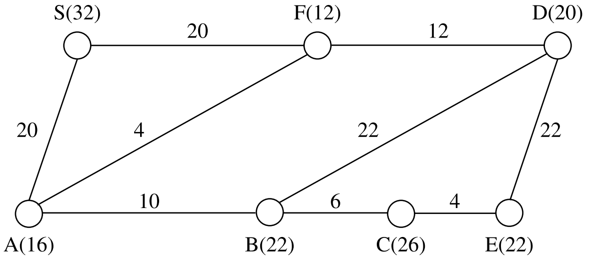


Figure 2. An example for Rule E1. The number on each edge represents the cost of the link between two nodes and the number in parenthesis presents the shortest path cost of the corresponding node to the given destination D.

2.4 Rule E2

Rule E2 is designed to extend deflection Rule 3, which eliminates immediate backtracking. Compared to Rule E1, Rule E2 can provide some complement to enlarge the successor set.

Rule E2: Given the destination node d, the successor set S_i at node i consists of neighbours $i+1$ for which node $i+1$ is not equal to node $i-1$ and satisfies any of the following:

$$D_{i+1}^d(G\backslash l_{i+1}) < D_i^d(G\backslash l_i), \tag{4}$$

$$D_{i+1}^d(G\backslash l_{i+1}) < D_{i-1}^d(G\backslash l_{i-1}), \tag{5}$$

$$D_{i+1}^d(G\backslash l_{i+1}) < D_{i-2}^d(G). \tag{6}$$

Theorem 2: All the paths discovered by Rule E2 between any pair of nodes are loop-free and deadend-free.

Proof: Removing links from the graph G increases the shortest path cost between two nodes. Therefore, for any node i and j ($i > 0, j > 0$), we have $D_i^d(G) \leq D_i^d(G\backslash l_j)$.

According to Rule E2

$$\begin{aligned} D_{i+1}^d(G\backslash l_{i+1}) &< \max\left\{D_i^d(G\backslash l_i), D_{i-1}^d(G\backslash l_{i-1}), D_{i-2}^d(G)\right\} \\ &\leq \max\left\{D_i^d(G\backslash l_i), D_{i-1}^d(G\backslash l_{i-1}), D_{i-2}^d(G\backslash l_{i-2})\right\}. \end{aligned}$$

To show that it is loop-free, we can use a similar method as in Rule E1. If we replace the G in Rule E1 with $G\backslash l_i$, Rule E1 and Rule E2 are of analogous form. Thus, by defining a similar two directional adjacent links cost, we can find that any two directional sequential links would never be visited more than twice in Rule E2.

To verify that it is deadend-free, we prove that the shortest-path neighbour in $G\backslash l_i$ is always an allowed choice. According to the definition of shortest-path neighbour $i+1$ in $G\backslash l_i$, $D_{i+1}^d(G\backslash l_i) < D_i^d(G\backslash l_i)$ and the shortest path of node $i+1$ to the given destination in $G\backslash l_i$ cannot go back across the last incoming link l_{i+1}, thus $D_{i+1}^d(G\backslash\{l_i, l_{i+1}\}) = D_{i+1}^d(G\backslash l_i)$.

Based on the above discussion

$$D_{i+1}^d(G\backslash l_{i+1}) \leq D_{i+1}^d(G\backslash\{l_i, l_{i+1}\}) = D_{i+1}^d(G\backslash l_i) \quad < D_i^d(G\backslash l_i).$$

The shortest-path neighbour in $G\backslash l_i$ satisfies Rule E2. □

For instance, in Figure 3, node S is sending a packet to node D. By applying Rule E1, it is possible for the packet to reach node A. When A receives that packet, Rule E1 cannot take node B into the successor set. This is because $\text{cost}(G, B)$ is larger than $\max\{\text{cost}(G, A)$, $\text{cost}(G, C)$, $\text{cost}(G, E)\}$. However, node B satisfies Rule E2, since $\text{cost}(G\backslash l_B, B)$ is 12 and $\text{cost}(G\backslash l_C, C)$ is 15. Thus, node B can be taken into A's successor set. Finally, path (S, F, E, C, A, B, D) is discovered.

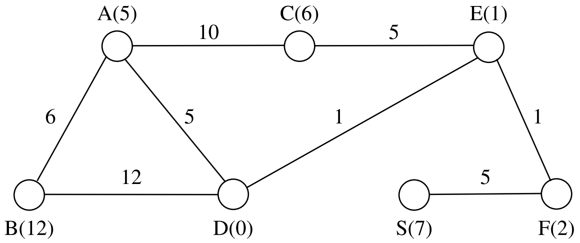


Figure 3. An example for Rule E2.

2.5 Relationship between Rule E1 and Rule E2

Let S_A denote the set of neighbours provided by Rule E1, S_B represents the set of neighbours satisfying Rule E2, and S_C is the intersection of set A and B. Obviously, nodes in set S_C satisfy one of the following conditions: (1) $D^d_{i+1}(G\backslash l_{i+1}) < D^d_i(G)$; (2) $D^d_{i+1}(G\backslash l_{i+1}) < D^d_{i-1}(G)$ and (3) $D^d_{i+1}(G\backslash l_{i+1}) < D^d_{i-2}(G)$. Note that set $S_B - S_C$ can be $\emptyset$, which means Rule E2 cannot always provide complement to Rule E1. Compared to Rule E1, (6) in Rule E2 cannot enrich the local successor set. This is because (6) in Rule E2 is a subset of (3) in Rule E1. The complement provided by Rule E2 is from its first two clauses (4) and (5), because $D^d_i(G) \leq D^d_i(G\backslash l_i)$ and $D^d_{i-1}(G) \leq D^d_{i-1}(G\backslash l_{i-1})$, which leads to a larger successor set. But when $D^d_i(G) = D^d_i(G\backslash l_i)$ and $D^d_{i-1}(G) = D^d_{i-1}(G\backslash l_{i-1})$, we can derive

$$D^d_{i+1}(G) < D^d_{i+1}(G\backslash l_{i+1}) < D^d_i(G\backslash l_i) = D^d_i(G),$$

$$D^d_{i+1}(G) < D^d_{i+1}(G\backslash l_{i+1}) < D^d_{i-1}(G\backslash l_{i-1}) = D^d_{i-1}(G).$$

Then, $S_B - S_C$ is $\emptyset$. In other words, given a destination, if the shortest path of node i does not include link l_i, and the shortest path of node $i-1$ does not include link l_{i-1}, Rule E1 will include all the nodes allowed by Rule E2. Otherwise, Rule E2 could take several additional neighbours in the successor set.

2.6 k-Hop history information based rules

The extended rules based on k-hop history information and one hop forward information ($k > 2$) is similar to Rule E1 and Rule E2. Since, Rule E1 and Rule E2 are almost the same except that a link is deleted in Rule E2, we only extend Rule E1 to show their properties.

Rule E1′: Given the destination node d, the successor set S_i at node i includes neighbours $i+1$ which satisfy either of the following conditions:

$$D^d_{i+1}(G) < D^d_i(G), \tag{7}$$

$$D^d_{i+1}(G) < D^d_{i-1}(G), \tag{8}$$

$$\cdots\cdots\cdots \tag{9}$$

$$D^d_{i+1}(G) < D^d_{i-k}(G). \tag{10}$$

We can prove that its loop-free and deadend-free with a similar method to the proof of Rule E1. The only difference is that we need to define the cost of k directional adjacent links instead of two directional adjacent links, which means we compare the cost in a $k + 1$ node group. Rule E1$'$ can take more neighbours in the successor set because it enlarges the qualified range of $D^d_{i+1}(G)$. However, in Rule E1$'$, one physical link may be visited $2k$ times in the worst case. Thus, while more history information can contribute to more diverse paths, it leads to unnecessary link revisit.

3. Routing based on successor set

Routing schemes based on successor sets are implemented by selecting a next hop from the successor set to forward packets at each passing node. We first give two next hop selection algorithms. One is *Random Selection*, while the other one is *Selection with Weight*. Then, we discuss successor set refinement to optimise our schemes.

3.1 Next hop selection algorithms

Selecting the next hop for forwarding incoming packets influences the path diversity and the path length. In this paper, we propose two next hop selection algorithms.

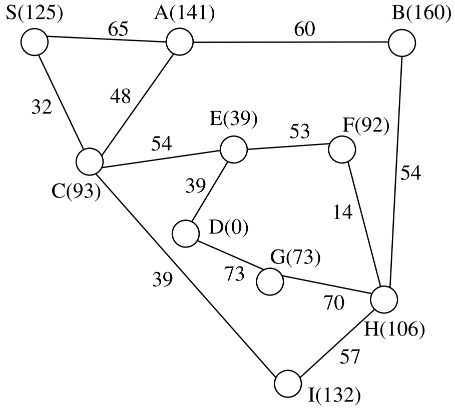
(A) Random selection. In the random selection scheme, a node randomly picks one element from its successor set as the next hop for delivering the incoming packet. This has three advantages: (1) random selection reduces the dependence on a single path or a single node; (2) most of the possible paths between a node pair can be utilised, which improves network utilisation and (3) it disperses traffic on different links so that data congestion on a single path is avoided.

Let λ denote the average size of successor set S, and *len* be the average length of all the discovered paths in terms of number of hops. The possibility $p(P_a)$ that a specified path P_a is taken is defined as $p(P_a) = \prod_{i=0}^{len}(1/\lambda)$. Thus, as the average size of the successor set and the average length of the paths increase, it is more likely to deliver packets on different paths. The corresponding algorithm is shown in Algorithm 1.

***Algorithm 1**. Random selection routing algorithm*

```
[1] Node constructs the successor set with Rule E1 or Rule E2 after receiving a packet
[2] Node randomly selects a node from the local successor set as the next hop
[3] Node forwards the packet to the chosen node in [2]
```

However, since our two rules focus on the loop-free and deadend-free properties without considering path length, random selection may lead to long paths. For instance, in Figure 4, node S is the source while D is the destination. Based on Rule E1, it is likely to take the path P_a:$(S, A, B, H, F, E, C, E, D)$ for packet routing, with latency 393. There are two drawbacks. First, path P_a is much longer than the shortest path (S, C, E, D) in terms of latency. In fact, when the packet reaches H, if H picks I as the next hop, the path can be even longer, which may exceed applications' tolerant latency. Second, it is more vulnerable to network failures. That is, if any nodes in $\{A, B, H, F, E, C\}$ fail, path P_a would be broken. Short paths in terms of number of hops are less fault-vulnerable under the same dynamic network. Based on the observation, we propose a better method.



Node	Successor Set
S	$\{A^*, C\}$
A	$\{C, B^*\}$
B	$\{H^*\}$
H	$\{F^*, I, G\}$
F	$\{E^*\}$
E	$\{C^*, D\}$
C	$\{E^*\}$
E	$\{D^*, F\}$
D	$\emptyset$

Figure 4. An example that random selection produces long paths. The left figure is the topology of a simple network. The corresponding successor set at each node for path $(S, A, B, H, F, E, C, E, D)$ is shown in the right table. The next hop chosen from the successor set is superscripted with a '$*$'.

(B) Selection with weight. In this approach, elements in the successor set are selected as the next hop with priority. Specifically, a node computes the weight of neighbours in the successor set according to their shortest path cost to the given destination. The smaller shortest-path cost they have, the larger weight they obtain. With the computed weight, a specified neighbour is selected as the next hop with probability.

Suppose λ_i is the size of the local successor set S_i at node i. r_j $(0 \leq j \leq \lambda_i)$ denotes the members in S_i. The weight of each neighbour in the successor set is presented by weight$_j$. We have the computation of weight$_j$ in S_i defined as follows:

$$\text{weight}_j = \frac{1/D^d_{r_j}(G)}{\sum_{j=0}^{\lambda_i}\left(1/D^d_{r_j}(G)\right)}, \tag{11}$$

or

$$\text{weight}_j = \frac{1/D^d_{r_j}(G \backslash l_{r_j})}{\sum_{j=0}^{\lambda_i}\left(1/D^d_{r_j}(G \backslash l_{r_j})\right)}. \tag{12}$$

(11) and (12) are for Rule E1 and Rule E2, respectively. At each node, $\sum_{j=0}^{\lambda_i}\text{weight}_j = 1$. We regulate the probability that a specified neighbour r_j in S_i is selected as the next hop is weight$_j$. Note that the weight calculation could be defined in other forms besides the reciprocal form. But the insight behind those two equations is that we want to select the nodes with smaller shortest-path cost as the next hop more frequently, which then results in that shorter paths are more likely to use. This is because the length of a path decreases when the shortest-path cost of the nodes along this path drops more quickly. As in Figure 4, node S tends to forward packets to node C instead of node A. C is more likely to choose node E as the next hop. Then, path (S, C, E, D) is utilised for data delivery, and path (S, A, B, H, I, C, E, D) is used as a backup. The above process is detailed in Algorithm 2.

There are several useful properties of this approach. First, it does not reduce the size of the successor set, which implies that compared to the random selection, the number of choices for routing from the local view and the number of diverse paths for packet delivery

Algorithm 2 *Selection with weight routing algorithm*

```
[1] Node constructs the successor set with Rule E1 or Rule E2 after receiving a packet
[2] Node computes the weight for each neighbour in the successor set
[3] Node selects one member from the local successor set as the next hop with probability and forwards the packet to it
```

Algorithm 3 *Successor set refinement algorithm*

```
[1] Node constructs the successor set using Rule E1 when it receives a packet
[2] Node computes $D^d_{i+1}(G\backslash l_{i+1})$ for each member in the successor set to find local deadend nodes
[3] Remove the local deadend nodes from the successor set
```

is not lessened. Second, with the difference of probability in selecting next hop, the average latency of the utilised paths decreases, without excluding any long paths. In addition, the average number of hops of the utilised paths is also reduced, which is proved by our simulations in Section IV. This allows that short paths (in terms of number of hops) are more frequently used for forwarding packets, which is less fault-vulnerable under the same dynamic network, while long paths are rarely utilised in practical. The reason that we do not completely exclude long paths here for routing is that the occasional utilisation of long paths could share the traffic load on the short paths. Moreover, long paths could be used as backup when short paths are found unhealthy (e.g., faulty and congested), and the transmission efficiency is improved.

3.2 Successor refinement

When Rule E1 applied, for improving path efficiency, some nodes should be removed from the successor set.

In Figure 5, according to Rule E1, path (S, A, E, F, E, D) is a valid choice between source S and destination D. In this case, when a packet reaches node F from node E, it has to come back to E in order to reach the destination D. We call nodes like F 'local deadend nodes'. A local deadend node only has one link in the network. When packets reach a local deadend node, they have no choice except returning. In Figure 5, there are two local deadend nodes: node C and node F. The local deadend node does not have any contribution to improve path diversity

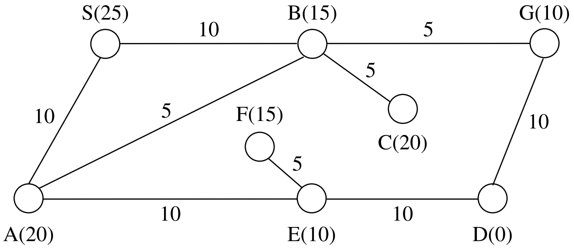


Figure 5. An example to illustrate the necessity of successor refinement.

because it cannot provide any other useful links for packet forwarding. In addition, choosing a local deadend node as the next hop increases the length of the path, which results in higher cost and latency. Thus, local deadend nodes should not be taken into the successor set.

To judge whether node $i+1$ is a local deadend node, we can compute its shortest path cost to the destination in graph $G\backslash l_{i+1}$, which is $D^d_{i+1}(G\backslash l_{i+1})$. If $D^d_{i+1}(G\backslash l_{i+1})$ is infinite, then node $i+1$ is a local deadend node and it would not be considered in the successor set. Note that according to the definition of our rules, local deadend nodes may only appear in Rule E1. This is because in Rule E2 there is already an upper bound for $D^d_{i+1}(G\backslash l_{i+1})$. For example, in Figure 5, when a packet reaches node E, E first computes the successor set by applying Rule E1, which is $\{A, F, D\}$. Then, it computes $D^d_{i+1}(G\backslash l_{i+1})$ for nodes A, F, D, respectively, and removes node F from the successor set since $D^d_F(G\backslash EF)$ is infinite. The algorithm is described in Algorithm 3.

4. Performance simulation

In this section, we evaluate SMRS and contrast its performance with the rules in [16]. Since in [16] Rule 2 includes all the choices allowed by Rule 1, our simulations only include Rule 2 and Rule 3.

4.1 Evaluation methodology

To show how SMRS performs in different network environments, we use GT-ITM [17] generating three topologies, with 128, 256 and 256 nodes, respectively. The model we used is the Waxman model [18], which generates graphs with a probability of an edge from one node to another. The average node degree in these three topologies is varied by tuning α and β. Specifically, an increase in α will raise the number of edges in the graph, while an increase in β will increase the ratio of long edges relative to shorter edges. Our objective is to evaluate the performance under the different network size scenario (Topology 1 and Topology 2) and the different average node degree scenario (Topology 2 and Topology 3). Link delay is used as the cost metric for routing in our simulation. Table 2 shows the detailed network topology generating parameters.

For each setup topology, we randomly choose 50 node pairs as the source and destination. Then we apply all four rules for routing. For each rule, every source runs 1024 times by sending packets to a destination, and we report the average. When utilising Rule E1, we apply the successor set refinement for optimisation.

To measure the performance of SMRS, we use the following metrics. (1) *Successor set size* is defined as the number of members in the successor set; (2) *Path diversity* is the number of disjoint paths between two nodes; (3) *Delivery ratio* records the ratio of the number of the connected pairs to the number of the sample pairs when some node failures occur in the network, which represents the ability to route around faults and (4) *Path length* is used to evaluate path

Table 2. Topology generating parameters in GT-ITM.

	Topology 1	*Topology 2*	*Topology 3*
Nodes	128	256	256
Average node degrees	16	16	19.5
Network scale	100 × 100	100 × 100	100 × 100
α	0.1	0.05	0.6
β	1	1	1

quality for each rule. We record the average latency and the average number of hops of paths between a node pair.

4.2 Performance results

We first evaluate the successor set size for each rule. Successor set size at a node, representing the number of choices for the next hop selection from the local view, is determined by the number of neighbours of that node, the successor set construction rule, the incoming packet, and the destination. Figure 6 records the percentages of the nodes with different successor set size in three topologies, where we contrast SMRS and deflection rules. To make it clear, we only plot the nodes whose successor set size is larger than 2. It shows that for any successor set size larger than 2, nodes with SMRS rules are always more than those with deflection rules. This is because with the route information of one more history hop, nodes can take more neighbours as the next hop to forward packets without loops.

In addition, according to our results, SMRS rules discover more different routes than the deflection rules. The reason is obvious. As the successor set size becomes larger at each node, which means there are more choices for the next hop selection at every passing node, the number of different routes increases. However, the number of different routes cannot present the path diversity precisely due to the shared nodes.

To better measure the path diversity, we define the correlation among the paths first. Let $P_a = (S, N_1, N_2, \ldots, N_n, D)$ and $P_b = (S, M_1, M_2, \ldots, M_m, D)$ be the two paths between source S and destination D, where $N_1, N_2, \ldots, N_n, M_1, M_2, \ldots, M_m$ are intermediate nodes and $n, m \geq 1$. S_p denotes the joint node set which contains the intermediate nodes shared by P_a and P_b. size(S_p) represents the number of nodes in S_p. The correlation of P_a and P_b, corr(P_a, P_b), is defined as $\mathrm{corr}(P_a, P_b) = \max\{(\mathrm{size}(S_p)/n), (\mathrm{size}(S_p)/m)\}$, and $0 \leq \mathrm{corr}(P_a, P_b) \leq 1$. Larger corr($P_a, P_b$) indicates more immediate nodes P_a and P_b share. If the shared immediate nodes fail, neither of these two paths work any longer. Thus, to corr(P_a, P_b), the smaller the better. If corr(P_a, P_b) is 0, P_a and P_b are two totally disjoint paths. For each node pair, we summarise the number of disjoint paths, which means that the correlation of any two paths is 0. This metric shows two important properties: (1) the ability to construct diverse paths of the successor set, we focus on whether larger successor set can provide more path diversity, (2) the fault-tolerance capacity, because the more disjoint paths there are, the more reliable it is under network faults. The first row in Figure 7 shows a comparison of the number of disjoint paths. SMRS rules provide more disjoint paths than deflection rules, which indicates that SMRS rules can provide more path diversity for routing.

Next, we evaluate the ability of routing under network faults. First, 50 different node pairs are randomly chosen in the network. The two nodes in a pair act as the source and destination, respectively. Then, we run the simulation 50 times. For the Nth time, we randomly select N

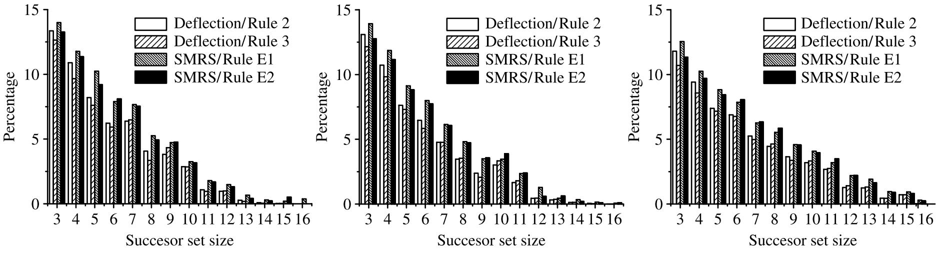


Figure 6. Comparison of successor set size. Left, Topology1; centre, Topology 2 and right, Topology 3.

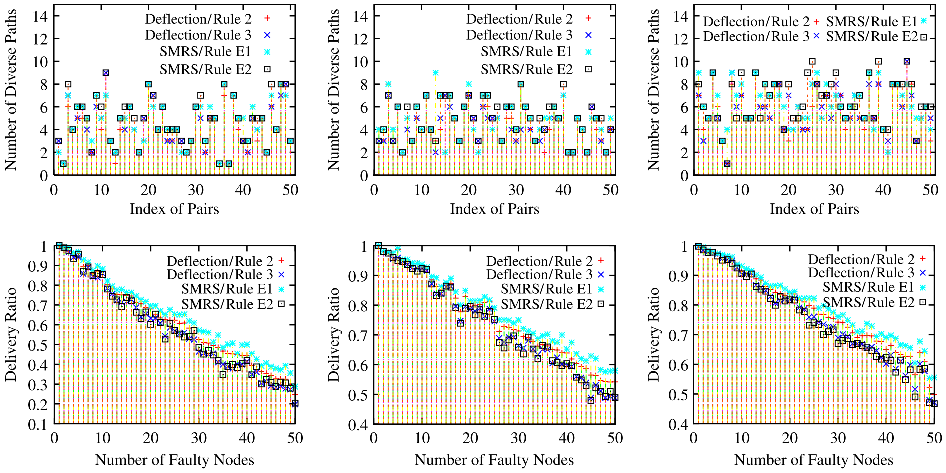


Figure 7. Comparison of number of disjoint paths (first row) and delivery ratio under network faults (second row). Left, Topology1; centre, Topology 2 and right, Topology 3.

nodes as faulty and let the source send packets to the destination 32 times with different rules applied. We consider the pair whose destination can receive the packets from the source as connected. The delivery ratio is defined as the ratio of the number of connected pairs to the number of sample pairs. The second row in Figure 7 summarises the delivery ratio for each rule. In Topology 1, when the number of faulty nodes is close to 50, SMRS rules can connect more pairs of nodes. In Topology 2 and Topology 3, the number of connected pairs is almost the same for SMRS rules and deflection rules. This is because when the network size (in Topology 2) or the average node degree (in Topology 3) grows, the impact of the network faults in Topology 1 on routing is reduced.

Then, we measure the performance of random selection and selection with weight. Figure 8 reports the effect of random selection and selection with weight schemes on path length, where

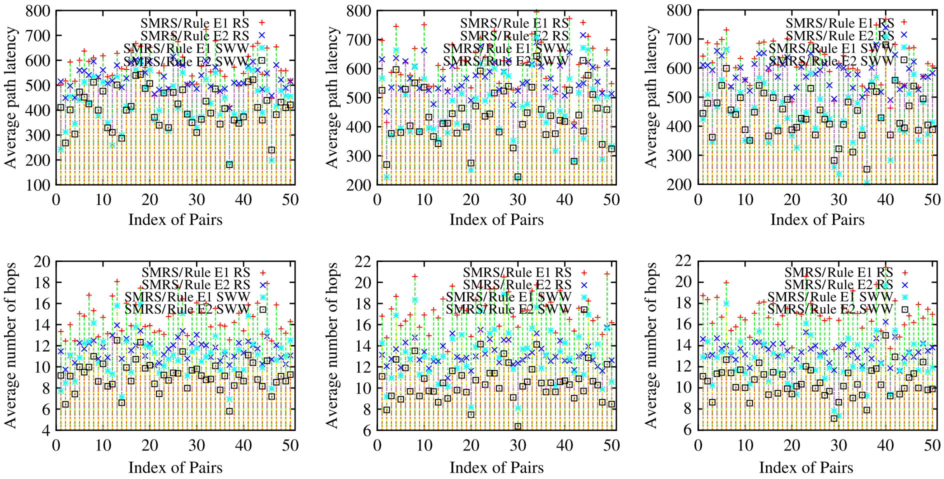


Figure 8. Comparison of length of paths in terms of latency (first row) and length of paths in terms of number of hops (second row). Left, Topology1; center, Topology 2 and right, Topology 3.

"RS" stands for random selection and 'SWW' is short for selection with weight. We measure the average latency and average number of hops of paths constructed by each rule between each node pair. The first row in Figure 8 presents the average latency of paths. We see that selection with weight decreases the average path latency by raising the probability of utilising short paths. The second row in Figure 8 shows the comparison of path length in terms of number of hops. It shows that although selection with weight is designed to reduce the average latency of paths, it also decreases the average number of hops compared to the random selection scheme. For all three networks, random selection leads to paths with a larger number of hops, which implies that routing with the random selection is more subject to network failures, although it can guarantee that all the paths are loop-free and deadend-free.

4.3 Evaluating Rule E1′ and Rule E2′

We then evaluate the performance of Rule E1′ and Rule E2′, which are the *k*-hop history information based rules. Specifically, we extend Rule E1′ and Rule E2′ based on *k*-hop history information, where *k* ranges from 1 to 6. Then, we randomly choose 50 node pairs as the source and destination, and apply random selection and selection with weight schemes for routing, respectively. For each rule, the source sends 1024 packets to the destination, and the average is reported.

The first row in Figure 9 plots the average length of paths in terms of number of hops as *k* varies. When *k* rises, the average number of hops increases. This is due to the fact that more history information causes more choices in the local successor set which may include neighbours far away from the destination. An interesting observation is that when $k > 2$, the average number of hops increases more quickly.

The second row in Figure 9 shows the number of different paths rules provide. We see that as network size grows and average node degree increases, both Rule E1′ and Rule E2′ discover more paths. Moreover, in all three topologies, the number of paths is much larger when $k \geq 2$ than when $k = 1$. Furthermore, the number of paths goes up slowly as *k* rises when $k \geq 2$. This implies that to discover more possible diverse loop-free and deadend-free paths with less

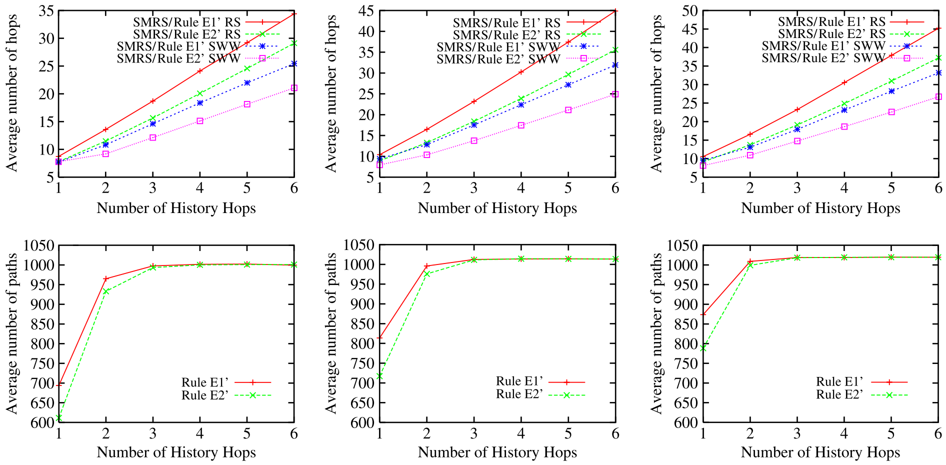


Figure 9. Comparison of length of paths in terms of number of hops (first row) and number of paths (second row). Left, Topology1; centre, Topology 2 and right, Topology 3.

knowledge, $k = 2$ is a relatively good choice, and 2-hop history information based rules outperform other rules.

5. Related work

Previous research has been conducted on multi-path routing [3,14] to improve Internet routing performance, especially under network failures. Source routing [5,13], as one of the approaches, has been explored extensively. In [15], Yang aimed at providing end users the ability to choose the sequence of Internet service providers a packet traverses. Raghavan et al. [8] presented an authenticated source routing system built around the concept of network capabilities. The issue of scaling to global internet sizes without restricting the availability of a diverse set of routes was addressed in [4]. The Nimrod architecture [2] was designed to accommodate a dynamic internetwork of arbitrary size with heterogeneous service, such as multi-path routing. In [19], routing decisions were made based on dynamic information, which was discovered by the routers with feedback and probes. However, all of those schemes are similar in that they utilise a link-state like routing to acquire the knowledge of the whole network for implementing source routing, which limits the scalability.

To address the above problem, per-hop routing schemes were proposed. In per-hop routing, instead of determining the complete path by source, each node along the path only determines the next hop of the packet with its local route information. In other words, the source node neither knows the whole network topology nor stores any instruction information in the forwarded packet, which improves scalability. OSPF [6,7] and ISIS [1,9] implemented multi-path routing through routing among next hops with equal cost to a given destination [10]. Vutukury et al. regulated that a router can forward packets to any neighbour with a cost less than itself for multi-path routing [11,12]. However, our approach can discover more diverse paths. Yang and Wetherall formulated three deflection set rules to qualify several neighbours for packet forwarding, which finally constructed a set of non-shortest paths to the destination without loops [16]. Our approach improves their work in three aspects. (1) SMRS rules construct more diverse loop-free and deadend-free paths with only one more history hop information; (2) algorithms are proposed to control the length of utilised paths and (3) SMRS aims at providing a set of routes diverse from each other, while the tag-based deflection discovers routes diverse from the shortest path.

6. Conclusion

In this paper, we have proposed a scalable multi-path routing scheme (SMRS), which consists of two components. One is the extension of the Yang and Wetherall deflection rules with one more history hop information to provide a set of successors for packet routing. The other is the next hop selection algorithms to control the average length of the utilised paths. Our simulations show that SMRS not only increases the number of choices for the next hop from the local view, but also discovers more diverse loop-free paths with less latency compared with the Yang and Wetherall approach. In addition, our simulations show that 2-hop history information based rules outperform other rules. For our future work, we plan to develop node degree-related protocols, for dynamically determining k in different network environments to improve multi-path routing performance with less overhead.

Acknowledgements

This work was supported in part by NSF Grants CCR 0329741, CNS 0422762, CNS 0434533, CNS 0531410 and CNS 0626240.

Note

1. Email: jie@cse.fau.edu

References

[1] R. Callon, *Use of OSI IS-IS for Routing in TCP/IP and Dual Environments*, RFC-1195, December 1990.

[2] I. Castineyra, N. Chiappa, and M. Steenstrup, *The nimrod routing architecture*, IETF RFC, 1992, August 1996.

[3] I. Cidon, R. Rom, and Y. Shavitt, *Analysis of multi-path routing*, IEEE/ACM Trans. Netw. 7(6) (1999), pp. 885–896.

[4] D. Estrin, Y. Rekhter, and S. Hotz, *Scalable inter-domain routing architecture*, in *Proceedings of ACM SIGCOMM*, 1992.

[5] Y. Ganjali and A. Keshavarzian, *Load balancing in ad hoc networks: single-path routing vs. multi-path routing*, in *Proceedings of IEEE Infocom*, 2004.

[6] C. Huitema, *Routing in the Internet*, 2nd ed., Prentice Hall, PTR, 2000.

[7] J. Moy, *OSPF version 2. Internet Draft RFC-2178*, July 1997.

[8] B. Raghavan and A.C. Snoeren, *A system for authenticated policy-compliant routing*, in *Proceedings of ACM SIGCOMM*, 2004.

[9] M. Steenstrup, *Routing in Communications Networks*, Prentice Hall, 1995.

[10] D. Thaler and C. Hopps, *Multipath issues in unicast and multicast next-hop selection*, in *IETF RFC 2991*, November 2000.

[11] S. Vutukury and J.J. Garcia-Luna-Aceves, *MDVA: A distance-vector multipath routing protocol*, in *Proceedings of IEEE Infocom*, 2001.

[12] ———, *MPATH: A loop-free multipath routing algorithm*, J. Microprocessors Microsyst. (2001).

[13] W. Wei and A. Zakhor, *Robust multipath source routing protocol (RMPSR) for video communication over wireless ad hoc networks*, in *Proceedings of ICME*, 2004.

[14] W. Xu and J. Rexford, *MIRO: Multi-path interdomain routing*, in *Proceedings of ACM SIGCOMM*, 2006.

[15] X. Yang, *NIRA: A new internet routing architecture*, in *Proc. ACM SIGCOMM FDNA workshop* (2003).

[16] X. Yang and D. Wetherall, *Source selectable path diversity via routing deflections*, in *Proceedings of ACM SIGCOMM*, 2006.

[17] K. Calvert and E. Zegura, *GT-ITM: Georgia Tech Internetwork Topology Models*, http://www.cc.gatech.edu/fac/Ellen.Zegura/graphs.html, 1997.

[18] E.W. Zegura, K.L. Calvert, and S. Bhattacharjee, *How to model an internetwork*, in *Proceedings of IEEE Infocom*, 1996.

[19] D. Zhu, M. Gritter, and D.R. Cheriton, *Feedback based routing*, in *Proceedings of HotNets-I*, 2002.