Partial Probing for Scaling Overlay Routing

Deke Guo, *Member*, *IEEE*, Hai Jin, *Senior Member*, *IEEE*, Tao Chen, *Member*, *IEEE*, Jie Wu, *Fellow*, *IEEE*, Li Lu, *Member*, *IEEE*, Dongsheng Li, *Member*, *IEEE*, and Xiaolei Zhou, *Student Member*, *IEEE*

Abstract—Recent work has demonstrated that path diversity is an effective way to improve the end-to-end performance of network applications. For every node pair in a full-mesh network with n nodes, this paper presents a family of new approaches that efficiently identify an acceptable indirect path that has a similar to or even better performance than the direct path, hence considerably scaling the network at the cost of low per-node traffic overhead. In prior techniques, every node frequently incurs $O(n^{1.5})$ traffic overhead to probe the links from itself to all other nodes and to broadcast its probing results to a small set of nodes. In contrast, in our approaches, each node measures its links to only $O(\sqrt{n})$ other nodes and transmits the measuring results to $O(\sqrt{n})$ other nodes, where the two node sets of size $O(\sqrt{n})$ are determined by the partial sampling schemes presented in this paper. Mathematical analyses and trace-driven simulations show that our approaches dramatically reduce the per-node traffic overhead to $O(n)$ while maintaining an acceptable backup path for each node pair with high probability. More precisely, our approaches, which are based on enhanced and rotational partial sampling schemes, are capable of increasing said probability to about 65 and 85 percent, respectively. For many network applications, this is sufficiently high such that the increased scalability outweighs such a drawback. In addition, it is not desirable to identify an outstanding backup path for every node pair in reality, due to the variable link quality.

Index Terms—Partial sampling, overlay network, backup path, scalability

1 Introduction

Recent research efforts [1], [2], [3] have demonstrated the potential of path diversity as an effective way to improve the end-to-end performance of network applications [4], [5]. This requires that a backup path should be available to take over in the presence of failure or a significant performance reduction on the default direct path. The current network infrastructure does not intrinsically support multipath routing. The diverse paths, however, can be obtained through an overlay network [6], [7], which can be used directly or can act as the backbone network in many applications [5], [8], [9]. In this setting, every pair of nodes requires an acceptable backup path that exhibits a good end-to-end performance while traversing additional relay nodes.

To address such an issue, conventional approaches make every node periodically monitor its links to other nodes and disseminate its link state table of $n-1$ entries to the others, where n is the number of nodes in the overlay [8], [10]. Consequently, every node is aware of the link state tables of all other nodes, making them all capable of periodically finding the best backup path for each node pair in the overlay network. Such approaches generate $O(n^2)$ per-node probing and disseminating overhead. They have been improved by reducing the traffic overhead to $O(n^{1.5})$ when every node exchanges its link state table with only $O(\sqrt{n})$ nodes. The improved approach ensures that there exists at least one rendezvous node that receives the link state tables from both members of every node pair. The best one among $n-2$ indirect paths between any node pair can, thus, be identified as the backup path for the node pair [5] by the rendezvous node.

Despite such progress, distributed algorithms that identify acceptable backup paths among all pairs of nodes remain a significant obstacle when scaling the network due to the following reasons. First, prior approaches make every node monitor the rest of the nodes frequently. The probing capability of every node, however, has practical limits due to the constraints in the link capacity and computation capability. Thus, every node would not monitor its links to too many other nodes in reality. In addition, the amount of overhead introduced into the network, due to the frequent per-node probing, is also considerably large. These two practical issues demonstrate that all-pairs probing only make sense in relatively small networks. Second, the size of the link state table at every node grows linearly with the number of probed nodes. As a result, with all-pairs probing, the frequent per-node dissemination of its link state table results in a large traffic overhead, especially for large-scale networks. In summary, having each node continuously monitor all of the other nodes is neither feasible nor desirable for large-scale networks.

- *D. Guo, T. Chen, and X. Zhou are with the Key Laboratory for Information System Engineering, National University of Defense Technology, Changsha 410073, P.R. China. E-mail: {guodeke, emilchenn, xlzhou.nudt}@gmail.com.*
- *H. Jin is with the SCTS&CGCL, School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, P.R. China. E-mail: hjin@hust.edu.cn.*
- *J. Wu is with the Department of Computer and Information Sciences, Temple University, 1805 N. Borad Street, Philadelphia, PA 19122. E-mail: jiewu@temple.edu.*
- *L. Lu is with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, P.R. China. E-mail: lulirui@gmail.com.*
- *D. Li is with the National Lab for Parallel and Distributed Processing, National University of Defense Technology, Changsha 410073, P.R. China. E-mail: dsli.lee@gmail.com.*

Manuscript received 17 Jan. 2012; revised 25 Oct. 2012; accepted 6 Nov. 2012; published online 28 Nov. 2012.
Recommended for acceptance by L. Xiao.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-2012-01-0031.
Digital Object Identifier no. 10.1109/TPDS.2012.326.

In this paper, we focus on the problem of identifying an acceptable backup path for every node pair in a full-mesh overlay network with as little per-node probing and disseminating overhead as possible; hence, we significantly scale the network. We tackle such a problem by presenting novel approaches that are based on partial sampling schemes for link-state routing in overlay networks. Our approaches operate in a two round mechanism: every node measures its links to $O(\sqrt{n})$ other nodes and then disseminates its link state table to $O(\sqrt{n})$ other nodes. As a result, our approaches incur $O(n)$ per-node probing and disseminating overhead, while the lowest per-node overhead is $O(n^{1.5})$ before our proposals.

In reality, such a problem brings about some challenges, which are presented in detail in Section 3.1. First, how can every node independently select a set of $O(\sqrt{n})$ other nodes to probe such that any node pair will finally probe some common nodes, each of which acts as a relay node? Second, how can every node select $O(\sqrt{n})$ other nodes to deliver its link state table to? In this way, at least one rendezvous node is aware of the link state tables of any node pair and can discover those alternative indirect paths for that node pair. Third, how can we infer as many alternative paths as possible for each node pair, given the very limited measuring results of that node pair? An acceptable backup path with a similar to or even better performance than the direct path can, thus, be identified.

To answer these issues, we formalize the first two challenges as the partial sampling problem and present its construction method. For the third issue, the associated path selecting approach can discover the best backup path from about $2\sqrt{n}$ alternative paths for every node pair in a distributed manner. We then present the enhanced partial sampling scheme and its path selecting approach, which generates about $6\sqrt{n}$ alternative paths for every node pair. Although such an enhanced scheme is a considerable improvement over the original one, some node pairs might need more alternative paths to discover the better backup path. We, thus, introduce the rotational partial sampling scheme to significantly improve the performance of each selected backup path from the fundamental way.

The experimental results show that our approaches that are based on the partial sampling scheme and its two variants, significantly reduce the resulting traffic overhead and support nearly $\sqrt{n}$ times as many nodes as prior approaches. Additionally, our approaches outperform the random approach [6] and the enhanced earliest-divergence approach [11] in terms of the probability that every recommended backup path has a similar to or even better performance than the direct path. Such a probability is about 65 percent for the enhanced partial sampling. Based on the enhanced partial sampling, the rotational partial sampling increases that probability to about 85 percent. For many network applications, this is sufficiently high such that the increased network scalability outweighs the drawback. It is worth noticing that such a probability can be further improved if more historical measuring results are utilized.

The rest of this paper is organized as follows: Section 2 summarizes the most related work. Section 3 presents the partial sampling scheme and the associated path selecting approach. In Section 4, we present the enhanced and rotational partial sampling schemes, and then we propose two associated path selecting approaches. Section 5 presents the performance evaluation results. We conclude this work in Section 6.

2 Related Work

Consider that many real-time applications have been deployed in the Internet, such as voice over IP [12], online video games, and so on. One fundamental requirement of such kinds of applications is the low delay between any pair of communicating nodes. However, the default path between any two nodes is not guided by the Internet and suffers failure and performance reduction in many cases. Many studies have reported the existence of triangle inequality violations (TIV) in the Internet delay space [13], [14], [15]. That is, it is possible to find a node C such that: $RTT(A,B) > RTT(A,C) + RTT(C,B)$, where $RTT(X,Y)$ denotes the round trip time between nodes X and Y. In this case, each node pair has a backup path with the node C as a relay node to take over the communication when the default path fails or exhibits a high delay.

Some novel methods have been proposed recently in different contexts [5], [8], [10], [16], [17] for identifying a backup path when establishing the default path between any node pair. One factor that restricts their wide use is the scalability limitations, which are due to the large amount of traffic overhead introduced into the network, involving link probing and link-state disseminating. Therefore, any reduction of said traffic overhead provides an opportunity to scale the network to more nodes. The per-node traffic overhead is, thus, reduced from $O(n^2)$ to $O(n^{1.5})$ in literature [5]. One of the main goals of our work is to significantly enhance the network scalability by reducing the per-node overhead to $O(\sqrt{n})$.

The one-hop source routing approach (only one relay node is utilized) was proposed in SOSR [6] to find an indirect path when recovering from Internet path failures. Every source node, however, is unaware of which intermediate relay node can provide a good backup path for reaching a given destination. Their experimental results demonstrate that having every source node randomly choose $k = 4$ intermediaries is enough to find a working backup path when recovering from a failed path. Each source sends packets through all k intermediaries in parallel and then routes through the intermediary whose response packet is first returned. Thus, the per-node traffic overhead is $O(k)$. We will show that the approach does not work well if the backup path should experience a similar to or even better performance than the default path.

The extended earliest-divergence rule in [11] assumes that every source node A knows the round-trip latency from itself to the destination node B, denoted as D_{AB}, and from itself to any relay node O, denoted as D_{AO}. The rule in [11] uses $D_{AO} + D_{AB}$ as an estimation for D_{OB} because the source node A is unaware of the latency between the relay node O and the destination node B. The source node A can infer the overall latency $D_{AOB} = 2 \times D_{AO} + D_{AB}$ of every indirect path to the destination node, and can randomly select one from the best m indirect paths according to the

estimated value of D_{AOB}. Thus, the per-node traffic overhead is $O(n)$. We will show that the approach does not perform well because every backup path does not have a good end-to-end performance with high probability.

3 Path Selecting Based on the Partial Sampling Scheme

We present a novel approach for finding an acceptable backup path for each node pair at the cost of every node only being able to probe $O(\sqrt{n})$ nodes and deliver its link state table to $O(\sqrt{n})$ nodes. We start by formalizing the problem as the partial sampling and propose the path selecting approach accordingly.

3.1 Problem Formulation

This paper tries to find a good backup path for each node pair in the network with as little per-node probing and disseminating as possible, thus significantly scaling the network. We, however, face three challenges as follows:

Although it is desirable to have each node only probe a small set of nodes, the first challenge is that every node is unaware of which nodes it should probe. This imposes a constraint where the intersection of two probing sets has to be nonempty for each node pair. A common node in two probing sets acts as a relay node and incurs one basic alternative path for that node pair. An intrinsic method for each node is to randomly select a small set of nodes from all n nodes. In this way, any two random probing sets have one common element with a given probability, where the size of each probing set is $O(\sqrt{n})$ [18]. In reality, such a probabilistic method suffers an obstacle; the intersection of two probing sets might be empty for some node pairs. Furthermore, the number of alternative paths is too small (at most two on average) to find one backup path, which outperforms the direct path of each node pair. Thus, a deterministic approach that provides more alternative paths for each node pair is desirable for this setting.

The second challenge involves selecting a set of nodes to deliver the link state table of every node to, which has measured its link states to a small set of nodes. This imposes a constraint where, for each node pair, both members send their link state tables to at least one common rendezvous node that can easily find those basic alternative paths. The centralized approach, where every node sends its link state table to a central node, suffers a single point of failure and a performance bottleneck. Those random approaches are also not suitable due to a similar reason as the one mentioned above. Therefore, distributed but deterministic approaches with less per-node traffic overhead are essential for this setting.

Once the above two challenges are addressed, each node pair can find the best backup path from those basic alternative paths, each with one relay node. In practice, some node pairs may need more alternative paths to discover the better backup path. The third challenge involves finding more alternative paths for each node pair without increasing the size of every probing set, so as to identify an acceptable backup path with a similar to or even better performance than the direct path.

The basic idea of our strategy to address the three challenges is characterized as Definition 1. Note that the first two challenges are the same in nature and can be represented by the first condition of Definition 1, while the third challenge is addressed by the second condition.

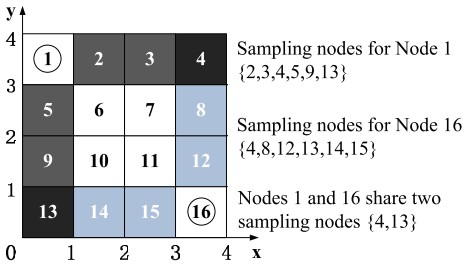


Fig. 1. An example of the partial sampling scheme.

Definition 1 (Partial Sampling). *For a set $Q = \{q_1, q_2, \ldots, q_n\}$ of n nodes, let $S(q_i)$ denote a set of α elements sampled from the set by q_i, where $1 \leq i \leq n$. Each node in $S(q_i)$ is selected by q_i for latency measuring. If all of the below conditions are satisfied, this scheme is called a partial sampling:*

1. *It holds that $S(q_i)$ and $S(q_j)$ have at least β common elements for any $i \neq j$, where $1 \leq i, j \leq n$.*
2. *It holds that, for each $x \in S(q_i)$, there exists an element $y \in S(q_j)$ such that $x \in S(y)$ and $y \in S(x)$, where $1 \leq i, j \leq n$.*
3. *For any pair of different nodes, q_i and q_j, in Q, the total number of sampling sets containing q_i is similar to that containing q_j, and is appropriately equal to α.*

The first condition demonstrates that, for each node pair, the number of β relay nodes are probed by both the source and the destination nodes. Thus, there exist β basic alternative paths for that node pair, which might be insufficient for finding a good backup path for the direct path. Therefore, the introduction of the second condition produces an opportunity to increase the total number of alternative paths for each node pair by providing some additional paths that traverse two relay nodes. As our results will show, some of these additional paths may exhibit the same even lower latencies than the direct path. Additionally, they are more powerful than the β alternatives when it comes to routing around failures over the direct Internet path. Actually, certain ISP policy constraints force nodes to take indirect paths with two relay nodes in order to route around failures [5].

If all nodes probe the same set of α intermediate nodes, it will intuitively generate $\beta = \alpha$ alternative paths with one relay node for each node pair. Although such a method satisfies the first two conditions, it causes imbalanced probing because only α nodes are probed by all of the nodes while others have never been probed. Thus, many better, indirect, alternative paths remain undiscovered. Therefore, the third condition arises to restrict the sampling scheme, derived from the first two conditions. Every node will be appropriately probed by α nodes in each round of recommending backup paths.

3.2 Construction of Partial Sampling

The construction method of the partial sampling is the key to realizing the motivation of this paper. One efficient way is to use the grid quorum systems [19], as shown in Fig. 1. A grid of size $\sqrt{n} \times \sqrt{n}$ contains n cells, each of which has a

unique identifier ranging from 1 to n and is filled with the n nodes of $Q = \{q_1, q_2, \ldots, q_n\}$ in any order. Without loss of generality, we assume that the node q_i fills the ith cell in the grid. We prefer to uniquely map all overlay nodes to the grid in a managed manner, i.e., there is a manager node in the overlay network. For any node q_i in position (x_i, y_i), let $S(q_i)$ denote a grid quorum that consists of $\alpha = 2\sqrt{n} - 2$ nodes in row x_i or column y_i, excluding itself. For another node q_j in position (x_j, y_j), $S(q_i)$ and $S(q_j)$ share two nodes in positions (x_i, y_j) and (x_j, y_i) if they are in different rows and columns; otherwise, they share $\sqrt{n} - 2$ nodes in the same row or column, excluding themselves.

This construction provides the following important properties and can implement the partial sampling scheme:

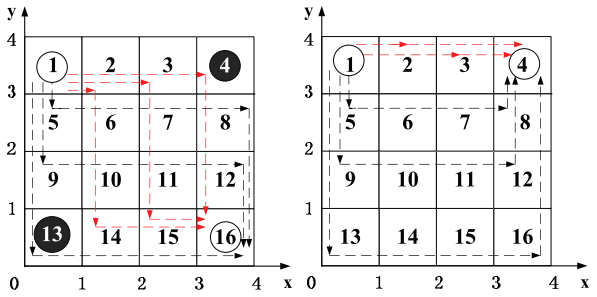
1. For each node pair, q_i and q_j, their sampling sets, $S(q_i)$ and $S(q_j)$, share $\beta = 2$ or $\sqrt{n} - 2$ common elements. Therefore, this approach provides $\beta = 2$ or $\sqrt{n} - 2$ alternative paths for the direct path between q_i and q_j.
2. For every node $x \in S(q_i)$, there exists a node $y \in S(q_j)$ such that nodes x and y are in the same row or column, and hence they probe each other. Thus, the second condition of Definition 1 holds. As discussed later, this incurs more alternative paths for the node pair q_i and q_j. Note that each of these additional paths traverses two relay nodes.
3. The probing load is evenly distributed among the nodes in the network. That is, every node q_i is probed by $2\sqrt{n} - 2$ nodes in its partial sampling set $S(q_i)$.

3.3 Backup Path Selection with the Partial Sampling

The above construction scheme needs efficient implementation approaches in reality. Such approaches should involve three basic stages. First, each node measures its link states to all of the other nodes in its partial sampling set and, thus, forms its link state table. Second, every node propagates its link state table to some rendezvous nodes. That is, those nodes receiving the probing results from node q_i are called the rendezvous nodes of q_i. Third, a common rendezvous node identifies one backup path for each node pair if it receives the link state tables from both members of that node pair.

A low-overhead approach would be when every node sends its probing results to a central rendezvous node, hence consuming $O(\sqrt{n})$ bandwidth per-node. This rendezvous node is responsible for calculating the latencies of possible alternative paths and selecting the one with the lowest latency as the backup path for each node pair in the network. This approach, however, suffers a single point of failure and a performance bottleneck. Another approach would be for every node to broadcast its partial probing results to all other nodes, hence consuming $O(n^{1.5})$ bandwidth per-node. This approach, however, provides more information than necessary such that every node becomes a common rendezvous node to calculate the backup path for each node pair.

To address these problems, our efficient strategy would be for every node q_i to send its partial probing results to all nodes in its partial sampling set $S(q_i)$. In this way, every node q_i acts as a rendezvous node for the other $2\sqrt{n} - 2$ nodes and maintains the probing results of such nodes. It is worth noticing that the partial sampling scheme is constructed so that the partial sampling sets of any two nodes share at least $\beta = 2$ nodes. Therefore, for any two different nodes, q_i and q_j, they have $\beta = 2$ rendezvous nodes in common.



(a) From node 1 to node 16. (b) From node 1 to node 4.

Fig. 2. An illustrative example of the alternative paths for the direct path when the network size is a perfect square.

Our algorithm then operates to identify the backup path from node q_i to node q_j. If nodes q_i and q_j are not in the same row or column, one common rendezvous node of q_i and q_j in position (x_j, y_i) computes the best one among the $2\sqrt{n} - 2$ alternative paths from q_i to q_j. Such paths include the number of $\sqrt{n} - 1$ paths (q_i, q_a, q'_a, q_j) and the number of $\sqrt{n} - 1$ paths (q_i, q_b, q'_b, q_j), where the relay nodes q_a, q'_a, q_b, and q'_b are in positions (a, y_i), (a, y_j), (x_i, b), and (x_j, b), respectively. Here, $a \in \{1, 2, \ldots, \sqrt{n}\} - \{x_i\}$, and $b \in \{1, 2, \ldots, \sqrt{n}\} - \{y_i\}$.

This computation can be performed by the common rendezvous node at position (x_j, y_i). Such a rendezvous node is aware of the link state tables of all nodes in its partial sampling set and knows the latency of each one-hop component of paths (q_i, q_a, q'_a, q_j) and (q_i, q_b, q'_b, q_j). For example, consider an alternative path (1, 2, 14, 16) from node 1 to node 16, as shown in Fig. 2a. The link state information of (1, 2) and (2, 14) has been reported to the common rendezvous node 4 by node 2, while that of (14, 16) has been reported by node 16. If we use the path (1, 5, 8, 16) as an example, the link state information of (1, 5) has been reported by node 1, while that of (5, 8) and (8, 16) has been reported by node 8.

Consequently, the common rendezvous node can calculate the latency for each of those $2\sqrt{n} - 2$ alternative paths from q_i to q_j, hence finding the best backup path among them. Finally, this rendezvous node sends the decision to nodes q_i and q_j. Note that another common rendezvous node of nodes q_i and q_j is in position (x_i, y_j), which always operates in the same way as the first one. As shown in Fig. 2a, nodes 4 and 13 are two common rendezvous nodes of nodes 1 and 16.

If nodes q_i and q_j are in the same row or column, our algorithm operates as follows: Since node q_i has received the link state tables of all nodes in its partial probing set $S(q_i)$, it can first compute the latencies of $\sqrt{n} - 2$ alternative paths from itself to node q_j locally. Such indirect paths are denoted as (q_i, q_a, q_j), where the relay node q_a can be any node in the same row or column with q_i and q_j, but not q_i and q_j. Additionally, node q_i can locally compute the

latencies of other $\sqrt{n}-1$ alternative paths (q_i, q_a, q_b, q_j) to node q_j. If q_i and q_j are in the same row, q_a and q_b are in positions (x_i, a) and (x_j, a), for $a \in \{1, 2, \ldots, \sqrt{n}\} - \{y_i\}$, respectively. Otherwise, q_a and q_b are in positions (a, y_i) and (a, y_j) for $a \in \{1, 2, \ldots, \sqrt{n}\} - \{x_i\}$, respectively. Fig. 2b gives an example of all alternative paths between nodes 1 and 4. In this way, every node q_i can find the best one among $2\sqrt{n}-3$ alternative paths to node q_j according to its local information.

This computation can be locally done because node q_i knows the link state tables of all nodes in its partial sampling set $S(q_i)$; thus, it knows the latency of each one-hop component of paths (q_i, q_a, q_b, q_j). For example, consider an alternative path (1, 13, 16, 4) from node 1 to node 4, as shown in Fig. 2b. The link state information of (1, 13) and (13, 16) has been reported to node 1 by node 13, while that of (16, 4) has been reported by node 4 to node 1. In the case of path (1, 3, 4), node 1 has probed the link state information of (1, 3) itself and knows the information of (3, 4) from nodes 3 and 4.

According to the above construction process of alternative paths for each node pair, we can derive Corollary 1.

Corollary 1. *For any node pair, the path selecting approach, based on the partial sampling, delivers $2\sqrt{n}-3$ alternative paths for the direct path if the pair members are in the same row or column; otherwise, $2\sqrt{n}-2$ alternative paths are given.*

To summarize, our efficient strategy where every node sends its partial probing results to all nodes in its partial sampling set provides enough information to identify an acceptable backup path for each node pair in the network. This is ensured by the two round operations at every node q_i. In the first round, node q_i identifies the backup path for each of those node pairs whose members are in its partial sampling set $S(q_i)$, but not in the same row or column. For any node pair whose members are in the set $S(q_i)$ and in the same row or column, their backup path can be locally identified by themselves. Furthermore, node q_i sends a recommendation message to every node q_j in $S(q_i)$ of $2\sqrt{n}-2$ nodes. Here, each message contains the information about those selected backup paths from node q_j to the other $\sqrt{n}-1$ nodes. In the second round, node q_i identifies the backup path from itself to every node in the set $S(q_i)$ locally.

We use Theorem 1 to measure the amount of per-node bandwidth consumption required to find the backup path for each node pair in the network.

Theorem 1. *This algorithm finds the backup path for each node pair in the network at the cost of each node generating at most $6\sqrt{n}$ messages and $O(n)$ bytes.*

Proof. The algorithm follows three steps as follows: In the first step, every node q_i measures the latencies on its paths to all nodes in its partial sampling set $S(q_i)$. This generates $2\sqrt{n}-2$ messages, each of which is a constant size, for example, 8 bytes for the ping operation, and hence incurs network traffic of $16(\sqrt{n}-1)$ bytes. In this way, node q_i can construct its link state table with $2\sqrt{n}-2$ entries, each of which uses 2 bytes for latency, 1 byte for liveness and loss, and 2 bytes for every node ID. Thus, the link state table of every node is $10(\sqrt{n}-1)$ bytes in size. In the second step, every node q_i sends its link state table to all nodes in $S(q_i)$ and results in $2(\sqrt{n}-1)$ messages for a total size of $20(\sqrt{n}-1)^2$ bytes. In the third step, every node q_i sends routing recommendations to all $2(\sqrt{n}-1)$ nodes in $S(q_i)$, where each recommendation consists of $\sqrt{n}-1$ entries. Here, each entry uses 2 bytes for the ID of the destination node and 4 bytes for, at most, two relay nodes. Thus, every node q_i generates $12(\sqrt{n}-1)^2$ bytes of network traffic in the third step.

In summary, every node causes $6(\sqrt{n}-1)$ total messages and $32n-48\sqrt{n}+16$ bytes so as to derive the backup path for each node pair in the network. □

4 Path Selecting Based on Enhanced Partial Sampling Scheme

We start with enhanced partial sampling and the associated path selecting approach to considerably improve the performance of the backup path for each node pair. We then present rotational partial sampling to improve the performance of each backup path from the fundamental way.

4.1 Problem Formulation

The partial sampling and associated path selecting approach in Section 3 can offer each node pair the best backup path among about $2\sqrt{n}-2$ alternative ones. The performance of the backup path for each node pair, however, can be considerably improved by tackling the following intrinsic limits of this approach. The first one is that the number of alternative paths between each node pair might be insufficient for identifying a desired backup path. That is, the third challenge mentioned in Section 3.1 arises. The second one is that every node always measures the same set of nodes and hence may omit some potentially better ones. These two limitations motivate us to explore a new path selecting approach.

The foundation of our new approach is *enhanced partial sampling*, which is just like partial sampling except we release the second condition of Definition 1. For each node pair, q_i and q_j, let $Es(q_i)$ and $Es(q_j)$ denote their enhanced partial sampling sets, respectively. It is not necessary for every node in $Es(q_i)$ to have a corresponding node in $Es(q_j)$ such that they sample each other, but the released second condition must be satisfied:

1. For every $x \in Es(q_i)$, the intersection of $Es(x)$ and $Es(q_j)$ is nonempty.
2. For every $x \in Es(q_j)$, the intersection of $Es(x)$ and $Es(q_i)$ is nonempty.

It is this released condition that brings about additional alternative paths for each node pair.

We present Definition 2 as an efficient construction method for the *enhanced partial sampling*, it is based on a grid of size $\sqrt{n}\times\sqrt{n}$, which is formed by using the method mentioned in Section 3. Fig. 3 gives an example of the enhanced partial sampling for a network with $n=25$ nodes.

Definition 2. *For every node q_i in position (x_i, y_i), $Es(q_i, k)$ is defined as the enhanced partial sampling set of q_i for any integer $1 \le k \le \sqrt{n}$. $Es(q_i, k)$ consists of all nodes in row $x_i^+(k)$ or column $y_i^+(k)$, where*

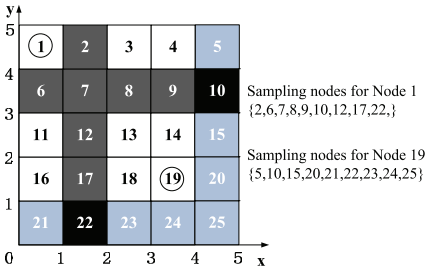


Fig. 3. An example of the enhanced partial sampling.

$$x_i^+(k) = \begin{cases} x_i + k, \text{ if } x_i + k \le \sqrt{n}, \\ x_i + k - \sqrt{n}, \text{ otherwise}, \end{cases} \quad (1)$$

$$y_i^+(k) = \begin{cases} y_i - k, \text{ if } y_i - k \ge 1, \\ y_i - k + \sqrt{n}, \text{ otherwise}. \end{cases} \quad (2)$$

Therefore, the size of $Es(q_i, k)$ is $\alpha = 2\sqrt{n} - 1$. We also define $x_i^-(k)$ and $y_i^-(k)$ as the reverse operation of $x_i^+(k)$ and $y_i^+(k)$, respectively.

Note that, for every node q_i, its enhanced partial sampling set $Es(q_i, k)$ can be implemented in $\sqrt{n}$ different ways, each with one possible value of k. Without loss of generality, we assume that $k = 1$, and we simplify the notations $Es(q_i, 1)$, $x_i^+(1)$, $y_i^+(1)$, $x_i^-(1)$, and $y_i^-(1)$ as $Es(q_i)$, x_i^+, y_i^+, x_i^-, and y_i^- in the remainder of this paper.

As we will show, the above construction method of $Es(q_i)$ for every node q_i in position (x_i, y_i) indeed satisfies the three conditions of enhanced partial sampling:

1. For any node q_j in position (x_j, y_j), $Es(q_i)$ and $Es(q_j)$ share two nodes in positions (x_i^+, y_j^+) and (x_j^+, y_i^+) if q_i and q_j are in different rows and columns. If q_i and q_j are in the same row with $y_i = y_j$, $Es(q_i)$ and $Es(q_j)$ share $\sqrt{n}$ nodes in the same row y_i^+. If q_i and q_j are in the same column, $Es(q_i)$ and $Es(q_j)$ share $\sqrt{n}$ nodes in the same column x_i^+. Therefore, this gives two or $\sqrt{n}$ alternative paths for the node pair q_i and q_j, each with one of the shared nodes as a relay node. Thus, the first condition is satisfied.
2. For every node $x \in Es(q_i)$, we can derive from the first condition that $Es(x)$ and $Es(q_j)$ share $\sqrt{n}$ nodes (if nodes x and q_j are in the same row or column) or two nodes. This generates $\sqrt{n}$ or two alternative paths from q_i to q_j with x and one common node in $Es(x)$ and $Es(q_j)$ as two relay nodes in order. Additionally, for every node $y \in Es(q_j)$, the same result holds for $Es(y)$ and $Es(q_i)$, hence producing $\sqrt{n}$ or two alternative paths from q_j to q_i through every y. Thus, the second condition is satisfied.
3. The entire probing load is evenly distributed among the nodes in the network, and hence, every node is probed by $2\sqrt{n} - 1$ nodes. Thus, the third condition is satisfied.

4.2 Backup Path Selection with the Enhanced Partial Sampling

The enhanced partial sampling can potentially provide some more alternative paths for each node pair in the network. Its implementation involves three basic stages. In the first stage, every node q_i measures its link states to nodes in the set $Es(q_i)$ and forms its link state table whose size is the cardinality of $Es(q_i)$. To utilize the partial view of the network that is observed by every node to find the backup path for each node pair, the following two stages that are conducted at every node are essential. They are the dissemination of the measuring results and the decision-making regarding the backup path. Actually, they have the same functionalities as the last two stages in our original approach in Section 3.3, but they differ in the technical details due to the changed sampling scheme.

In the second stage, a straightforward method would be for every node q_i to send its link state table to all nodes in its enhanced partial sampling set $Es(q_i)$. In this way, for each node pair, q_i and q_j, there exist two or $\sqrt{n}$ common rendezvous nodes, each of which can identify the two or $\sqrt{n}$ alternative paths that result from the first condition of enhanced partial sampling. Such common rendezvous nodes, however, cannot find more alternative paths derived from the second condition because they are unaware of the link state table of every node $x \in Es(q_i)$. For example, for the two common rendezvous nodes 10 and 22 of nodes $q_i = 1$ and $q_j = 19$, node 10 only receives the link state table from node 2 among $Es(1)$, and node 22 only receives the same from node 6 among $Es(1)$ if every node only sends its link state table to the nodes in its enhanced partial sampling set, as shown in Fig. 3. Thus, this backup path selecting method meets the same limitation that is also faced by our prior approach based on partial sampling: there is an insufficient number of alternative paths for each node pair.

Therefore, a feasible method for this setting would be for every node q_i to send its link state table to all nodes in $Es(q_i)$ as well as $S(q_i)$. For the direct path from node q_i, in position (x_i, y_i), to node q_j, in position (x_j, y_j), this method ensures that at least one rendezvous node is aware of the link state tables of node q_i, node q_j, and all nodes in $Es(q_i)$. Thus, this rendezvous node can discover the best one of those alternative paths for each node pair, q_i and q_j, derived from the second condition of the enhanced partial sampling. We provide the details from the following three viewpoints. The selected path for the direct path from q_i to q_j is different from that of q_j to q_i, unless nodes q_i and q_j are in the same row or column.

If nodes q_i and q_j are in different rows and columns, the nodes in positions (x_j, y_i^+) and (x_j^+, y_i^+) are two common rendezvous nodes of q_i, q_j, and the nodes not only in $Es(q_i)$ but also in row y_i^+. The node in position (x_j^+, y_i^+) is a preferred one, while the node in position (x_j, y_i^+) is a redundant one. For example, as shown in Fig. 4, the left column demonstrates all alternative paths derived by the preferred common rendezvous node 10. The received link state tables by every such rendezvous node are sufficient enough to find the best one among $3\sqrt{n} - 3$ alternative paths as follows:

1. $2\sqrt{n} - 4$ paths (q_i, q_a, q_b, q_j), where node q_a is in position $(x_a, y_a = y_i^+)$, and node q_b is in positions (x_a^+, y_j^+) and (x_j^+, y_a^+). Here, $x_a \in \{1, 2, \ldots, \sqrt{n}\} - \{x_j, x_j^+\}$.

Decisions at the rendezvous node 10	Decisions at the rendezvous node 22
1→6→22→19; 1→6→15→19	1→2→23→19; 1→2→10→19
1→7→23→19; 1→7→15→19	1→7→23→19; 1→7→15→19
1→8→24→19; 1→8→15→19	1→12→23→19; 1→12→20→19
1→9→5→19; 1→9→10→19; 1→9→15→19; 1→9→20→19; 1→9→25→19	1→17→21→19; 1→17→22→19; 1→17→23→19; 1→17→24→19; 1→17→25→19
1→10→19	1→22→19

Fig. 4. Two common rendezvous nodes derive a total number of $6\sqrt{n} - 8$ distinct alternative paths for the direct path from node 1 to node 19.

Alternative paths from 1 to 4		Alternative paths from 1 to 16	
Decisions at node 1	1→6→4;1→7→4; 1→8→4;1→9→4; 1→10→4	Decisions at node 1	1→2→16;1→7→16; 1→12→16;1→17→16; 1→22→16
Decisions at node 7	1→2→10→4; 1→7→15→4; 1→12→20→4; 1→17→25→4; 1→22→5→4	Decisions at node 7	1→6→22→16; 1→7→23→16; 1→8→24→16; 1→9→25→16; 1→10→21→16

Fig. 5. The source and one common rendezvous nodes derive a total number of $2\sqrt{n}$ distinct alternative paths for any two nodes in the same row or column.

2. $\sqrt{n}$ paths (q_i, q_a, q_b, q_j), where the relay nodes q_a and q_b are in positions (x_j, y_i^+) and (x_j^+, y_b), respectively. Here, y_b ranges from 1 to $\sqrt{n}$.
3. The path from q_i to q_j, where the node in position (x_j^+, y_i^+) is a relay node.

In addition, nodes in positions (x_i^+, y_j) and (x_i^+, y_j^+) are two common rendezvous nodes of q_i, q_j, and the nodes in not only $Es(q_i)$ but also in column x_i^+. Although any such rendezvous node can calculate the best one among $3\sqrt{n} - 3$ alternative paths as follows, the node at position (x_i^+, y_j^+) is a preferred common rendezvous node, while another node acts as a redundant one. For example, as shown in Fig. 4, the right column demonstrates all alternative paths derived by the preferred common rendezvous node 22:

1. $2\sqrt{n} - 4$ paths (q_i, q_a, q_b, q_j), where node q_a is in position $(x_a = x_i^+, y_a)$, and node q_b is in positions (x_j^+, y_a^+) and (x_a^+, y_j^+). Here, $y_a \in \{1, 2, \ldots, \sqrt{n}\} - \{y_j, y_j^+\}$.
2. $\sqrt{n}$ paths (q_i, q_a, q_b, q_j), where the relay nodes q_a and q_b are in positions (x_i^+, y_j) and (x_b, y_j^+), respectively. Here, x_b ranges from 1 to $\sqrt{n}$.
3. The path from q_i to q_j, where the node in position (x_i^+, y_j^+) is a relay node.

In the case where q_i and q_j are in the same row, they are aware of each other's link state tables, resulting from the dissemination method of link measuring results at every node. As a result, the source node q_i can locally calculate the best one from $\sqrt{n}$ alternative paths, each with one node in row y_i^+ as the relay node. In addition, the node at position (x_i^+, y_i^+) is a common rendezvous node of q_i, q_j, and the nodes in $Es(q_i)$. Another node, either in position $(x_i^+ + 1, y_i^+)$ or $(x_i^+ + 1 - \sqrt{n}, y_i^+)$, acts as a redundant rendezvous node in common. Any such rendezvous node can calculate the best one among $\sqrt{n}$ alternative paths (q_i, q_a, q_b, q_j) for the direct path from q_i to q_j. Note that q_a and q_b are in positions (x_i^+, y_a) and (x_j^+, y_a^+), respectively, where $y_a \in \{1, 2, \ldots, \sqrt{n}\}$. In summary, there are $2\sqrt{n}$ alternative paths from q_i to q_j, for example, all alternative paths from 1 to 4 are shown in the left column in Fig. 5, where nodes 7 and 8 are common rendezvous nodes.

In the case where q_i and q_j are in the same column, the source node q_i can directly calculate the best one from $\sqrt{n}$ alternative paths, each with one node in column x_i^+ as the relay node. Furthermore, the node at position (x_i^+, y_i^+) is a common rendezvous node of q_i, q_j, and the nodes in $Es(q_i)$. Another node, either at position $(x_i^+, y_i^+ - 1)$ or $(x_i^+, y_i^+ - 1 + \sqrt{n})$, is a redundant rendezvous node in common. Every such common rendezvous node can find the best one among $\sqrt{n}$ alternative paths (q_i, q_a, q_b, q_j) for the direct path from q_i to q_j. Note that q_a and q_b are in positions (x_a, y_i^+) and (x_a^+, y_j^+), respectively, where $x_a \in \{1, 2, \ldots, \sqrt{n}\}$. In summary, there exist $2\sqrt{n}$ alternative paths from q_i to q_j. As an example of such a case, all alternative paths from 1 to 16 are demonstrated in the right column in Fig. 5, where nodes 1 and 12 are two rendezvous nodes in common.

Corollary 2. *Based on enhanced partial sampling, the backup path selecting approach delivers $6\sqrt{n} - 8$ or $2\sqrt{n}$ alternative paths for the direct path from q_i to q_j.*

Proof. When q_i and q_j are in different rows and columns, each of the two kinds of rendezvous nodes calculates $3\sqrt{n} - 3$ distinct alternative paths from q_i to q_j. Consider the fact that the two sets of alternative paths have two common paths, as shown in Fig. 5. Thus, the total number of distinct alternative paths is $6\sqrt{n} - 8$ for this setting. As aforementioned, the total number of alternative paths from q_i to q_j is $2\sqrt{n}$ when q_i and q_j are in the same row or column. □

We use Theorem 2 to summarize the basic idea of our new approach based on the enhanced partial sampling, and then we characterize the amount of per-node bandwidth consumption.

Theorem 2. *The approach based on enhanced partial sampling finds the backup path for each node pair with every node incurring at most $8\sqrt{n}$ total messages and $O(n)$ bytes.*

Proof. As mentioned above, every node performs three types of per-node communications, including probing its link states to other nodes, delivering its link state table, and responding with the selected backup paths. First of all, every node q_i at position (x_i, y_i) measures all nodes in its sampling set $Es(q_i)$ by the ping operation. More precisely, this generates $2\sqrt{n} - 1$ messages for a total size of $16\sqrt{n} - 8$ bytes. Thus, node q_i forms its link state table with $2\sqrt{n} - 1$ entries and has a total size of $10\sqrt{n} - 5$ bytes. Furthermore, every node q_i sends its link state table to $4\sqrt{n} - 5$ nodes in $S(q_i)$ and $Es(q_i)$, thus resulting in $4\sqrt{n} - 5$ messages for a total size of $40n - 70\sqrt{n} + 25$ bytes. This provides a sufficient amount of information to identify a good backup path for each node pair through three round operations at every node q_i.

In the first round, node q_i discovers the best backup path for each node pair, q_a and q_b, which are in row y_i^- or column x_i^-, but cannot be in the same row or the

same column. As a result, node q_i sends a recommendation message to each of the $2\sqrt{n}-2$ nodes in row y_i^- and column x_i^-, excluding the node at position (x_i^-, y_i^-), with each message consisting of $\sqrt{n}-1$ entries. Here, each entry uses 2 bytes for the ID of the destination node and 4 bytes for, at most, two relay nodes. Thus, every q_i generates $12(\sqrt{n}-1)^2$ bytes of network traffic in this round.

In the second round, node q_i discovers the best one among $\sqrt{n}$ alternative paths for the direct path from the node in position (x_i^-, y_i^-) to every other node in row y_i^- or column x_i^-. Consequently, node q_i sends one additional message of $2\sqrt{n}-2$ entries to the node at position (x_i^-, y_i^-), resulting in $12(\sqrt{n}-1)$ bytes of network traffic. In the third round, node q_i locally discovers the best one among $\sqrt{n}$ alternative paths for the direct path from itself to every node in the set $S(q_i)$ without causing any network traffic.

In summary, every node causes $8\sqrt{n}-7$ messages and $52n-66\sqrt{n}+17$ bytes so as to identify one good backup path for each node pair in the network. □

4.3 Rotational Partial Sampling

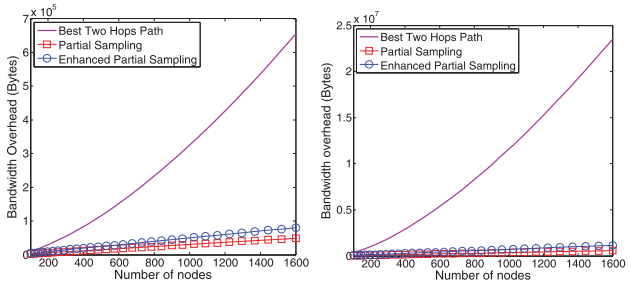
The enhanced partial sampling significantly increases the number of alternative paths for each node pair in the network. Every node, however, always measures the same set of nodes, result in missing some useful paths to other nodes. This motivates us to propose *rotational partial sampling*, which makes every node q_i probe a different set of nodes in each round, and all other nodes get probed by q_i after $\sqrt{n}$ rounds. At least one rendezvous node can observe the link states from every node q_i to the rest nodes in the network after $\sqrt{n}$ rounds.

The enhanced partial sampling that we present in Definition 2 can implement the motivation of *rotational partial sampling* in a natural way. The basic strategy would be for every node q_i to construct its partial sampling set as $Es(q_i, k)$, which varies as k increases (the round of sampling). The value of k is reset to 1 once it exceeds $\sqrt{n}$ because $Es(q_i, k_1) = Es(q_i, k_2)$ when $|k_2 - k_1|\%\sqrt{n} = 0$ for different k_1 and k_2. Thus, all other nodes will be probed by any node q_i every $\sqrt{n}$ rounds. After defining the partial sampling set for every node in a rotational manner, every node q_i measures all nodes in the set $Es(q_i, k)$ and sends its link state table to all nodes in $S(q_i)$ and $Es(q_i, k)$. In this way, the path selecting approach, based on enhanced partial sampling, can be adopted directly as the path selecting approach, based on rotational partial sampling, in each round.

At the same time, every node q_i achieves the entire view about its link states to the rest of the nodes in the network after $\sqrt{n}$ rounds. However, only about $1/\sqrt{n}$ of the observed view is refreshed while other parts become historical records. Therefore, the path selecting approach, based on rotational partial sampling, can exhibit a better performance if some parts of the historical measuring results of every node are utilized.

5 Evaluation

In this section, we start with two traces from real network systems. We then evaluate the performance of our approaches in finding an acceptable backup path for each node pair; we use in-system emulations based on the two traces.



(a) Probe other nodes using the ping operation. (b) Probe other nodes using the traceroute operation.

Fig. 6. Comparison of the average amount of network traffic incurred by per-node in each round.

5.1 Description of Data Sets

A. PlanetLab Trace. This trace shows the maximum, average, and minimum latencies between all node pairs on PlanetLab [20] from January 2004 to June 2005. The per-node probing/disseminating interval is 15 minute. A subset of this data set is exacted for our evaluations, which lasted from April 1, 2005 until April 4, 2005 in a scale of about 440 nodes.

B. iPlane Trace. The iPlane [21] service publishes the traceroute results from 200 source nodes to 140,000 destination nodes every day. All source nodes are PlanetLab nodes, and the destination nodes contain all source nodes. After collecting the iPlane trace from April 1, 2011 to May 30, 2011, we extracted an archive of traceroute between each node pair on 169 PlanetLab nodes for our evaluations. The per-node probing/disseminating interval is one day in such a trace.

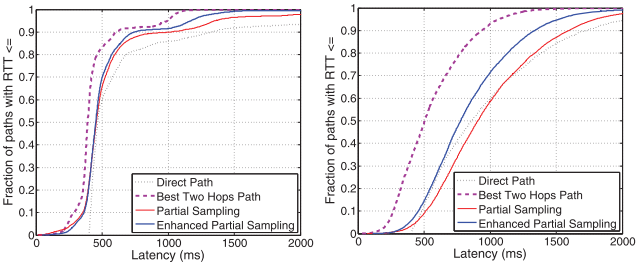
5.2 Overhead: Bandwidth Consumption

We first evaluate the per-node bandwidth consumption of our backup path selecting approaches and the selecting approach for the best two-hops path (it traverses two relay nodes) in [5], which is the best one before our proposals. Note that our approach that is based on the rotational partial probing scheme consumes the same bandwidth compared to that which is based on the enhanced partial sampling scheme. We perform the evaluation by using in-system emulations; under the first case, where every node probes other nodes using the ping operation, and under the second case using the traceroute operation. The experimental results are plotted in Fig. 6.

We can see that our approaches indeed dramatically reduce the per-node bandwidth consumption compared to prior approaches in [5], irrespective of the network size. This demonstrates that our approaches scale the network as expected.

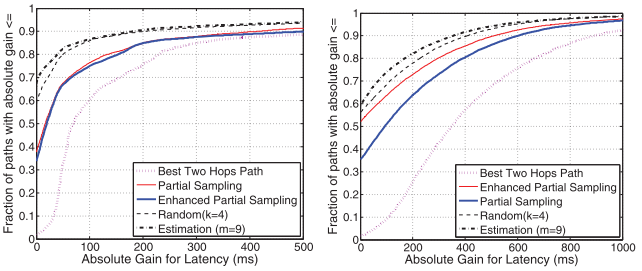
5.3 Effectiveness

We then perform a measurement study on the latencies of the direct Internet path and the indirect backup paths from five approaches for each node pair. They are the best two-hops path selecting approach in [5], the first two



(a) PlanetLab trace on April 4, 2005. (b) iPlane trace on May 30, 2011.

Fig. 7. Comparison of RTT for pairs of PlanetLab hosts whose point-to-point latencies are larger than 400 ms (high latency paths).



(a) PlanetLab trace on April 4, 2005. (b) iPlane trace on May 30, 2011.

Fig. 8. The absolute gain of latency on average, due to the introduction of the backup path for each node pair.

approaches presented in this paper, the random selection approach with $k = 4$ in [6], and the enhanced earliest-divergence approach with $m = 9$ in [11] (called the estimation approach here). Fig. 7 plots the CDF of path latency for different settings.

We first extract 9,241 pairs of communicating nodes from the PlanetLab trace whose end-to-end latencies along the direct Internet paths are larger than 400 ms. Fig. 7a shows the improvement in latency given by the best two-hops path and the backup path from our approaches for those 9,241 direct Internet paths. Fig. 7b shows that for 14,558 direct Internet paths whose point-to-point latencies are larger than 400 ms in the iPlane trace. We can see that our two approaches introduce a considerable improvement in latency compared to the direct Internet path despite its greatly reduced bandwidth consumption. This proves that the backup path with one or two relay nodes can exhibit less latency than the direct path for many node pairs. Additionally, our approach that is based on the enhanced partial sampling outperforms that which is based on the partial sampling, as was expected. The latencies given by the paths from the random selection approach and estimation approach are omitted in Fig. 7, and our approaches outperform them, as shown in Fig. 8.

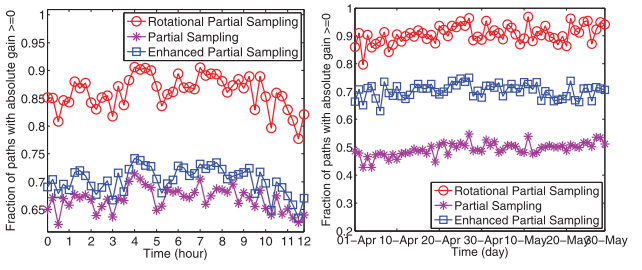
With these measured results, we compare our approaches with others in terms of the absolute gain. Here, the latency on the direct path minus the latency of the backup path, recommended by different approaches, is defined as the absolute gain. Fig. 8 plots the CDF of the absolute gain for the two different settings. We can see that the best two-hop path approach almost always finds a backup path exhibiting a lower latency than the direct path for each node pair. The root cause is that every node measures its links to all other nodes, and at least one common rendezvous node is aware of the latencies of all possible two-hop alternative paths for each node pair.

Additionally, our approaches can ensure that the recommended backup path exhibits the same even better end-to-end latency with a probability of about 65 percent in comparison to the latency of the direct path for each node pair. It is clear that our two approaches work better than the random selection approach and the estimation approach; however, we cannot achieve a similar performance to that of the best two-hop path approach. The root reason is that every node measures at most $2\sqrt{n}$ other nodes, and at most $6\sqrt{n}$ alternative paths can be identified for the direct path between each node pair. To improve the performance of each recommended backup path, we further propose the path selecting approach based on *rotational partial sampling*.

We evaluate our three path selecting approaches on the two traces over a relatively long period time. Fig. 9 shows that the new approach achieves a relatively stable improvement in the delay of selected backup paths over a long time period, even if it only uses the measuring results of every node during the current and last rounds. More precisely, for each node pair, the rotational partial sampling increases the probability that the resulting backup path has a similar or lower delay than the direct path to about 85 percent. Such a probability is usually sufficient and can be further increased if more historical measuring results are used, for example, the measuring results during the current and last two rounds. Additionally, it is not necessary to identify the best backup path for each node pair in each round because another backup path will be discovered from a different set of alternative paths in the next round. The probability that the recommended backup path for each node pair exhibits a lower performance than the default path in two continuous rounds is very low, just 2.25 percent.

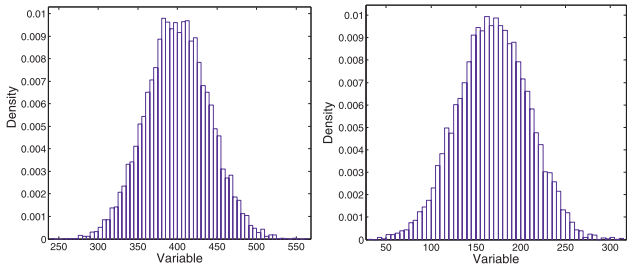
5.4 Discussion

Note that our motivation is to significantly scale the network by identifying an acceptable backup path for each node pair with as little per-node traffic overhead as possible. An acceptable backup path has a similar to or even better performance than the related direct path. More precisely, the partial and enhanced partial probing schemes considerably reduce the per-node traffic overhead from $(n^{1.5})$ in [5] to $O(n)$. As a tradeoff, the sparsity of such two partial probing schemes constrains to absolutely identify the best backup path for every node pair as prior work in [5] does, as shown in Figs. 7 and 8.



(a) PlanetLab trace lasting 12 hours on April 4, 2005. (b) iPlane trace lasting from April 1, 2011 to May 30, 2011.

Fig. 9. A comparison between approaches in their ability to pick backup paths, which have the same or an even better performance than the direct paths.



(a) PlanetLab trace lasting 12 hours on April 4, 2005.

(b) iPlane trace lasting from April 1, 2011 to May 30, 2011.

Fig. 10. The distribution of a variable that denotes the number of relay loads.

To compensate such a tradeoff, we propose the rotational partial sampling scheme where every node q_i probe a different set of nodes in each round, and all other nodes get probed by q_i after $\sqrt{n}$ rounds. Thus, at least one rendezvous node can collect the link states from node q_i to the rest nodes in the network after $\sqrt{n}$ rounds. Such a new scheme can identify the better backup paths compared to our prior schemes, even if it only uses the measuring results during the current and last rounds, as shown in Fig. 9. It is an extreme case if all measuring results of each node q_i during the past $\sqrt{n}$ rounds are utilized. In this setting, our rotational partial sampling scheme appears as the all-pairs ping method in [5]. The common rendezvous node of any node pair q_i and q_j can find the best backup path with only a relay node at a certain level of accuracy.

On the other hand, the historical measuring results by every node q_i can be utilized to derive some statistical models [22], [23], such as the path delay model. At the same time, all nodes in $S(q_i)$ keep the entire view about the link states from node q_i to all other nodes after $\sqrt{n}$ rounds; hence, they can also derive such statistical models as node q_i does. Such statistical models are complementary to our approaches because the delays of partial or whole unmeasured paths of each node can be predicted. This works well, especially for those paths that are not measured at the current round but have been measured recently. Thus, the predicted and measured link states of every node can be combined to provide more alternative paths and to discover a more outstanding backup path for each node pair. We leave such a research issue as one of our future work.

Moreover, each node may be chosen as the relay node by multiple node pairs. The number of node pairs that use a node as the relay node is a discrete variable. We can derive from Fig. 10 that the probability distribution of the variable is similar to the normal distribution under the two traces. Thus, some nodes may be chosen as the relay nodes by more node pairs than other nodes and it is likely that such attractive relay nodes may suffer from congestion in an overlay network. The proposed backup path selecting methods as well as the existing methods suffer such a common problem in an overlay network. Additionally, the selected backup path is only utilized in the presence of failure or a significant performance reduction on the direct path between any node pair. Thus, the real impact that such selected backup paths impose on the network traffic is not ease to evaluated during the distributed selection process of each individual backup path. For such constraints, it is better to tackle the potential traffic congestion via the existing traffic control techniques. More detailed discussion on such a problem, however, we leave as one of our future work.

6 Conclusion

Path diversity is an effective way to improve the end-to-end performance of network applications. In prior techniques in this setting, each node periodically introduces $O(n^{1.5})$ traffic overhead in the network. This paper proposes a family of new approaches, in which every node measures its links to $\sqrt{n}$ other nodes and transmits its measured results to $\sqrt{n}$ other nodes. This dramatically reduces the cost of per-node probing and disseminating to $O(n)$ while maintaining an acceptable backup path for each node pair, with a probability of about 85 percent. For many applications, this is sufficiently high such that the improved scalability of networks outweighs this drawback. We offer an exciting step in scaling full-mesh overlay networks.

We plan to study several issues in the future. The first issue is to study the mechanisms that ensure that our approaches continue to perform well in the presence of node and link failures. Second, we will redesign our approaches to find a backup path that is not heavily correlated with the direct path.

Acknowledgments

The authors would like to thank anonymous reviewers for their constructive comments. The work was partially supported by the National Basic Research Program (973 program) under Grant No. 2014CB347800, and the NSFC under Grants Nos. 61170284, 61202487, and 61222205.

References

[1] F. Wang and L. Gao, "A Backup Route Aware Routing Protocol-Fast Recovery from Transient Routing Failures," *Proc. IEEE 27th INFOCOM,* pp. 2333-2341, Apr. 2008.

[2] K.-W. Kwong, L. Gao, R. Guerin, and Z.-L. Zhang, "On the Feasibility and Efficacy of Protection Routing in IP Networks," *Proc. IEEE 29th INFOCOM,* Mar. 2010.

[3] F. Wang and L. Gao, "Path Diversity Aware Interdomain Routing," *Proc. IEEE 28th INFOCOM,* pp. 307-315, Apr. 2009.

[4] M. Jain and C. Dovrolis, "Path Selection Using Available Bandwidth Estimation in Overlay-Based Video Streaming," *Computer Networks,* vol. 52, no. 12, pp. 2411-2418, 2008.

[5] D. Sontag, Y. Zhang, A. Phanishayee, D.G. Andersen, and D. Karger, "Scaling All-Pairs Overlay Routing," *Proc. ACM Fifth Int'l Conf. Emerging Networking Experiments Technologies (CONEXT),* Sept. 2009.

[6] P.K. Gummadi, H.V. Madhyastha, S.D. Gribble, H.M. Levy, and D. Wetherall, "Improving the Reliability of Internet Paths with One-Hop Source Routing," *Proc. Sixth Conf. Symp. Opearting Systems Design and Implementation (OSDI),* pp. 183-198, Dec. 2004.

[7] D. Guo, J. Wu, Y. Liu, H. Jin, H. Chen, and T. Chen, "Quasi-Kautz Digraphs for Peer-to-Peer Networks," *IEEE Trans. Parallel Distributed Systems,* vol. 22, no. 6, pp. 1042-1055, June 2011.

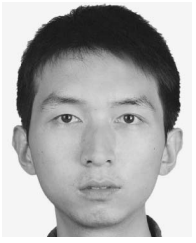
[8] A. Nakao, L.L. Peterson, and A.C. Bavier, "Scalable Routing Overlay Networks," *Operating Systems Rev.,* vol. 40, no. 1, pp. 49-61, 2006.

[9] R.S. Kazemzadeh and H.-A. Jacobsen, "Adaptive Multi-Path Publication Forwarding in the Publiy Distributed Publish/Subscribe Systems," technical report, Univ. of Toronto, Canada, Nov. 2011.

[10] D.G. Andersen, H. Balakrishnan, M.F. Kaashoek, and R. Morris, "Resilient Overlay Networks," *Proc. 18th ACM Symp. Operating Systems Principles (SOSP),* pp. 131-145, Oct. 2001.
[11] T. Fei, S. Tao, L. Gao, and R. Guérin, "How to Select a Good Alternate Path in Large Peer-to-Peer Systems?" *Proc. IEEE 25th INFOCOM,* Apr. 2006.
[12] H. Li, L. Mason, and M. Rabbat, "Distributed Adaptive Diverse Routing for Voice-over-IP in Service Overlay Networks," *IEEE Trans. Network and Service Management,* vol. 6, no. 3, pp. 175-189, Sept. 2009.
[13] G. Wang, B. Zhang, and T.S.E. Ng, "Towards Network Triangle Inequality Violation Aware Distributed Systems," *Proc. Seventh ACM SIGCOMM Conf. Internet Measurement (IMC),* pp. 175-188, Oct. 2007.
[14] C. Lumezanu, R. Baden, N. Spring, and B. Bhattacharjee, "Triangle Inequality Variations in the Internet," *Proc. Ninth ACM SIGCOMM Conf. Internet Measurement (IMC),* pp. 177-183, Nov. 2009.
[15] D.R. Choffnes, M.A. Sánchez, and F.E. Bustamante, "Network Positioning from the Edge - An Empirical Study of the Effectiveness of Network Positioning in P2P Systems," *Proc. IEEE 29th INFOCOM,* pp. 291-295, Mar. 2010.
[16] W. Cui, I. Stoica, and R.H. Katz, "Backup Path Allocation based on a Correlated Link Failure Probability Model in Overlay Networks," *Proc. IEEE 10th Int'l Conf. Network Protocols (ICNP),* Nov. 2002.
[17] F. Cantin, B. Gueye, D. Kaafar, and G. Leduc, "Overlay Routing Using Coordinate Systems," *Proc. ACM CONEXT Conf.,* 2008.
[18] W.W. Terpstra, J. Kangasharju, C. Leng, and A.P. Buchmann, "Bubblestorm: Resilient, Probabilistic, and Exhaustive Peer-to-Peer Search," *Proc. ACM SIGCOMM,* pp. 49-60, Aug. 2007.
[19] R. Friedman, G. Kliot, and C. Avin, "Probabilistic Quorum Systems in Wireless Ad Hoc Networks," *ACM Trans. Computer Systems,* vol. 28, no. 3, pp. 184-206, 2010.
[20] M. Olbrich, F. Nadolni, F. Idzikowski, and H. Woesner, "Measurements of Path Characteristics in Planetlab," Technical Report TKN-09-005, Technical Univ. Berlin, Berlin, July 2009.
[21] H. Madhyastha, E. Katz-Bassett, T. Anderson, A. Krishnamurthy, and A. Venkataramani, "iPlane Nano: Path Prediction for Peer-to-Peer Applications," *Proc. Sixth USENIX Symp. Networked Systems Design Implementation (NSDI),* pp. 137-152, 2009.
[22] B.D. Abrahao and R.D. Kleinberg, "On the Internet Delay Space Dimensionality," *Proc. Eighth ACM SIGCOMM Conf. Internet Measurement (IMC),* pp. 157-168, Oct. 2008.
[23] D.K. Lee, K. Jang, C. Lee, G. Iannaccone, and S.B. Moon, "Scalable and Systematic Internet-Wide Path and Delay Estimation from Existing Measurements," *Computer Networks,* vol. 55, no. 3, pp. 838-855, 2011.

Deke Guo received the BS degree in industry engineering from Beijing University of Aeronautic and Astronautic, Beijing, China, in 2001, and the PhD degree in management science and engineering from National University of Defense Technology, Changsha, China, in 2008. He was a visiting scholar at the Department of Computer Science and Engineering in Hong Kong University of Science and Technology from January 2007 to January 2009. He is an associate professor with the College of Information System and Management, National University of Defense Technology, Changsha, China. His research interests include distributed systems, wireless and mobile systems, P2P networks, and interconnection networks. He is a member of the ACM and the IEEE.

Hai Jin received the PhD degree in computer engineering in 1994 from the Huazhong University of Science and Technology (HUST), China, where he is currently the Cheung Kong professor and the dean of the School of Computer Science and Technology. In 1996, he received a German Academic Exchange Service fellowship to visit the Technical University of Chemnitz in Germany. He was at the University of Hong Kong between 1998 and 2000, and as a visiting scholar at the University of Southern California between 1999 and 2000. He received the Distinguished Young Scholar Award from the National Science Foundation of China in 2001. He is the chief scientist of ChinaGrid, the largest grid computing project in China. He is the member of Grid Forum Steering Group (GFSG). He has coauthored 15 books and published more than 400 research papers. His research interests include computer architecture, virtualization technology, cluster computing and grid computing, peer-to-peer computing, network storage, and network security. He is a senior member of the IEEE and a member of the ACM.



Tao Chen received the BS degree in military science, the MS and PhD degrees in military operational research from the National University of Defense Technology, Changsha, China, in 2004, 2006, and 2011, respectively. He is an assistant professor with the College of Information System and Management, National University of Defense Technology, Changsha, P.R. China. His research interests include wireless sensor networks, peer-to-peer computing, and data center networking. He is a member of the IEEE.

Jie Wu is the chair and a professor in the Department of Computer and Information Sciences, Temple University. Prior to joining Temple University, he was a program director at National Science Foundation. His research interests include wireless networks and mobile computing, routing protocols, fault-tolerant computing, and interconnection networks. He has published more than 500 papers in various journals and conference proceedings. He serves in the editorial board of the *IEEE Transactions on Computers*, *IEEE Transactions on Mobile Computing*, and *Journal of Parallel and Distributed Computing*. He is a program cochair for IEEE INFOCOM 2011. He was also a general cochair for IEEE MASS 2006, IEEE IPDPS 2008, ACM WiMD 2009, and IEEE/ACM DCOSS 2009. He also served as a panel chair for ACM MobiCom 2009. He has served as an IEEE computer society distinguished visitor. Currently, he is the chair of the IEEE Technical Committee on Distributed Processing (TCDP) and ACM distinguished speaker. He is a fellow of the IEEE.

Li Lu received the BS and MS degrees from Zhejiang University, China, in 2000 and 2003, respectively, all in electrical engineering. He received the PhD degree in computer science and engineering from Chinese Academy of Science in 2007. He is currently an associate professor in the School of Computer Science and Engineering at the University of Electronic Science and Technology of China. His research interests include RFID system, security in wireless networks, and applied cryptography. He is a member of the IEEE Computer Society, and a member of ACM. He is a member of the IEEE.

Dongsheng Li received the BS and PhD degrees in computer science from College of Computer, National University of Defense Technology, Changsha, China, in 1999 and 2005, respectively. He received the prize of National Excellent Doctoral Dissertation of P.R. China by Ministry of Education of China in 2008. He is currently an associate professor at National Lab for Parallel and Distributed Processing, National University of Defense Technology, China. His research interests include peer-to-peer computing, cloud computing, and computer network. He is a member of the IEEE.

Xiaolei Zhou received the BS degree in information management from Nanjing University, Nanjing, China, in 2009, and the MS degree in military science from National University of Defense Technology, China, in 2011. He is currently working toward the PhD degree in College of Information System and Management, National University of Defense Technology, China. His current research interests include wireless sensor networks, Internet of things, and data center networking. He is a student member of the IEEE.