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,* Xiaolei Zhou, *Member, IEEE*

Abstract—Recent work has demonstrated that path diversity is an effective way for improving 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 for efficiently identifying 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 analysis 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 every node pair with high probability. More precisely, our approaches which are based on the enhanced and rotational partial sampling schemes, would be capable of increasing said probability to about 65% and 85%, 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 absolutely 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

There is a growing demand for the Internet to provide reliable services as it carries more and more mission critical applications, such as voice over IP [1] (VoIP) and online games. One essential requirement of such kinds of applications is the low delay between any pair of communicating nodes, i.e, the end-to-end performance. Unfortunately, failures are fairly common in the Internet due to various causes. When such failures occur in the default path between any two nodes, an alternate path should be available to take over the default direct path.

Recent research efforts [2], [3], [4] have demonstrated the potential of path diversity as an effective way for improving the end-to-end performance of network applications [5], [6]. While the current network infrastructure does not intrinsically support multi-path routing, the diverse paths can be obtained through an overlay network [7], [8], which can be used directly or can act as the backbone network in many applications [6], [9], [10]. Furthermore, while it is possible to keep more alternate paths for every default path, this would incur considerable overhead. Therefore, every pair of communicating nodes usually identifies an acceptable backup path that traverses given relay nodes and exhibits a good end-to-end performance.

There is a challenge that arises here; although every node in the overlay can actively measure its links to all of the other nodes, for any node pair, A and B, no node is aware of the link status from both nodes A and B to any relay node D. Thus, the best backup path for any node pair cannot be identified by both members of that node pair or other nodes in the network.

To address such an issue, conventional approaches make every node not only periodically monitor its links to the rest of the nodes, but also disseminate its link state table of $n-1$ entries to the others, where n is the number of nodes in the overlay [9], [11]. Consequently, every node is aware of the link state tables of all other nodes, hence being capable of periodically finding the best backup path for each node pair in the overlay. Such approaches generate $O(n^2)$ per-node probing and disseminating overhead and 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 selected by the quorum system [12]. The improved approach ensures that for every node pair there exists at least one rendezvous node that receives the link state tables from both members of that node pair; hence, the best backup path from $n-2$ indirect paths is identified [6].

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. Firstly, the 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

- *D. Guo, T. Chen, and X. Zhou are with the Key laboratory for Information System Engineering, College of Information System and Management, 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.*
- *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.*

probing, is also considerably larger. These two practical issues demonstrate that all-pairs probing only makes sense in relatively small networks. Secondly, 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 large traffic overhead, especially for large-scale networks. In summary, having every node continuously monitors all of the other nodes is neither feasible nor desirable for large-scale networks.

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 as possible; hence, we significantly scale the network. We tackle such a problem by presenting novel approaches 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. For every node pair in the network, this mechanism ensures that at least one rendezvous node receives the link state tables from both members and can discover the best from about $6\sqrt{n}$ alternative paths for the direct path. 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 technical problem brings about some challenging issues, which are presented in detail in Section 3.1. Firstly, how can every node independently select a set of $O(\sqrt{n})$ other nodes from the network to probe such that any pair of nodes finally probe some common nodes, each of which acts as a relay node? Note that one alternative indirect path for that node pair is generated through one of such relay nodes. Secondly, how can every node select a set of $O(\sqrt{n})$ other nodes to deliver its link state table such that 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? Thirdly, how can we infer as many alternative paths as possible for every node pair, so as to identify an acceptable backup path with a similar to or even better performance than the direct path, given the very limited measuring results of that pair of nodes?

To answer these issues, we first formalize the first two issues as the partial sampling problem and present its construction method inspired by the quorum systems. 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. The number of alternative paths might be insufficient to contain an acceptable backup path for every node pair. Therefore, we present the enhanced partial sampling scheme and its path selecting approach, which can discover about $6\sqrt{n}$ alternative paths for every node pair at the cost of only increasing the size of the probing set by one at all nodes. Although this enhanced scheme achieves considerable improvement over the original one, some node pairs might need more alternative paths to discover the better backup path. Inspired by this fact, we further introduce the rotational partial sampling scheme for significantly improving the performance of each selected backup path from the fundamental way.

The experimental results show that our approaches which 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 those prior approaches. Additionally, our approaches outperform the random approach [7] and the enhanced earliest-divergence approach [13] in terms of the probability that every recommended backup path has a similar to or even better performance than the direct path. Actually, this probability is about 65% for the enhanced partial sampling. Based on the enhanced partial sampling, the rotational partial sampling increases that probability to about 85%, even if it only uses the measuring results of every node during the current and last rounds. For many network applications, this is sufficiently high such that the increased network scalability outweighs the drawback. It is worth noticing that such probability can be further improved if more historical measuring results of every node are used. Additionally, it is not necessary to absolutely identify an outstanding backup path for every node pair in each round since 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 every node pair exhibits lower performance than the default path in two continuous rounds is only 2.5% and is very low.

The rest of this paper is organized as follows. Section 2 summarizes the most related work. Section 3 presents the partial sampling scheme and 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 [1], online video games, etc. 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 such constraints on 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 [14], [15], [16]. That is, it is possible to find an intermediate 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, we intend to use node C as a relay node instead of sending the data directly from node A to node B. That is, each node pair can have a backup path to take over the communication when the default path fails or exhibits high delay.

A number of novel methods for identifying backup paths have been proposed recently in different contexts [17]. As pointed out in [18], these methods can be roughly divided into two categories. One is the reactive method, which does not reserve any backup path for the default path between any two nodes, hence initiating a search for a new path when the default

path fails [2], [3], [4], [7], [13]. Such methods suffer the non-trivial delay due to having to temporarily search for a new path; hence, they would not support many applications well, such as real time and streaming media applications. The other is the proactive method, which identifies at least one backup path when establishing the default path [6], [9], [11], [18] so as to accommodate the stringent performance requirements of applications. In this paper, we study the backup path routing in proactive restoration.

Despite the potential benefits and usages in many contexts, such as inside some distributed storage products, in route optimization products [6], and in distributed publish/subscribe systems [10], the proactive methods are difficult to utilize in large-scale systems. One factor restricting the wider use is their scalability limits, 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. For such a reason, the per-node traffic overhead is reduced from $O(n^2)$ to $O(n^{1.5})$ in literature [6]. One of the main goals of our work is to significantly enhance the network scalability by reducing the per-node overhead to $O(\sqrt{n})$. It is worth noticing that several technical details in [6], that involve implementing the grid quorum over an overlay network, are also suitable for the approaches presented in this paper after some modifications, such as the management and maintenance of the grid quorum.

The one-hop[1] source routing approach is used in SOSR [7] to find an indirect path for recovering from Internet path failures. One challenge that arises here is that every source node 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 the failed Internet path. Our results show that the approach works poorly if the backup path is required to experience a similar to or even better performance than the default path.

The extended earliest-divergence rule in [13] tries to identify backup paths that are as disjoint as possible from the default path. Although this rule is proposed at the AS level, it is actually also suitable at the IP level. It first assumes that every source node A is aware of 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 [13] then uses D_{AO}+D_{AB} as an estimation for D_{OB} since the source node A is unaware of the round-trip latency between the relay node O and the destination node B. As a result, the source node A can infer the overall latency D_{AOB}=2×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}. Our results show that the approach performs poorly since every backup path does not have a good end-to-end performance with high probability.

1. Actually, it means that only one relay node is utilized.

3 Path selects Based on Partial Sampling Scheme

We present a novel approach for finding an acceptable backup path for every 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 with formalizing the problem as partial sampling and propose the path selecting approach accordingly.

3.1 Problem formulation

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

Although it is desirable that every node only probes 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 every node pair. A common node in two probing sets acts as a relay node and incurs one basic alternative path for that node pair. The theory of the birthday paradox ensures that any two random probing sets have one element in common with a given probability, where the size of each probing set is $O(\sqrt{n})$ [19]. In reality, such probabilistic means suffer an obstacle; the intersection of two probing sets might be empty for some node pairs. Furthermore, the number of alternative paths is too low (at most two on average) to find one backup path, which outperforms the direct path of every node pair. Thus, a deterministic approach that brings more alternative paths for every 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, which has measured its link states to a small set of nodes. This imposes a constraint where, for every node pair, both members send their link state tables to at least one common rendezvous node, which 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 performance bottleneck. Those random approaches that are based on the theory of birthday paradox 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, every node pair can find the best backup path from those basic alternative paths, each with one relay node. In reality, the number of such basic alternative paths for every node pair is limited while more such paths require more common elements in any two probing sets, hence enlarging the sampling set of each node. Thus, the selected backup path for every node pair may not outperform the direct path in terms of latency. The third challenge that arises here involves finding more alternative paths for every 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

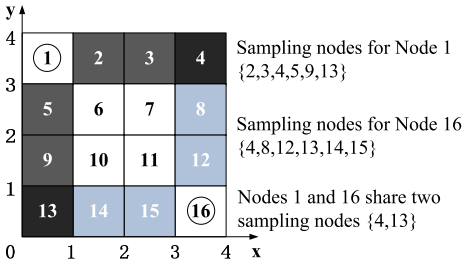


Fig. 1. An illustrative example of the partial sampling scheme.

the first condition of Definition 1, while the third challenge is addressed by the second condition.

Definition 1: **Partial Sampling** For a set $Q=\{q_1,q_2,...,q_n\}$ 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 every node pair, 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 gives an opportunity to increase the total number of alternative paths for every 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 [6].

If all nodes probe the same set of α intermediate nodes, it will intuitively generate $\beta=\alpha$ alternative paths with one relay node for every node pair. Although such a method satisfies the first two conditions, it causes imbalanced probing since only α nodes are probed by all of the nodes while others have never been probed; hence, many better, indirect, alternative paths remain undiscovered. Therefore, the third condition arises to restrict the sampling scheme derived from the first two conditions such that 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 for the partial sampling is the key to realizing the motivation of this paper. One efficient way is to use the grid quorum systems [12], 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 $Q=\{q_1,q_2,...,q_n\}$ in any order. Without loss of generality, we assume that the node q_i fills the i^{th} grid cell in a managed way. That is, there is a manager node in the overlay network. It is responsible to build a grid, allocate identifers to the overlay nodes, and map those overlay nodes to the grid cells.

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 except 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 except themselves.

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

1) Firstly, for every 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) Secondly, 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) Thirdly, 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 Selecting the backup path based on the partial sampling

The motivation behind our partial sampling is to find the acceptable backup path for every node pair in the network with as little per-node probing and disseminating as possible. Although the above construction scheme of partial sampling is feasible in theory, it needs efficient implementation approaches in reality. Such approaches should involve three basic stages. In the first stage, every node measures its links states to all of the other nodes in its partial sampling set and hence forms its link state table. In the second stage, 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. In the third stage, a common rendezvous node identifies one backup path for every node pair if it receives the link state tables from both members of that node pair.

One approach with low-overhead would be when every node disseminates 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 identifying 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 rendezvous node to calculate the backup path for every node pair.

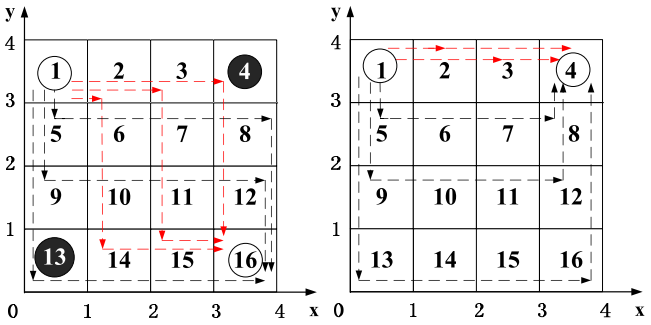
To address these problems, our efficient strategy would be for every node q_i to send its partial probing results to all nodes in its partial sampling set $S(q_i)$. In this way, every node q_i acts as a rendezvous node for the other $2\sqrt{n}-2$ nodes and hence maintains the probing results of such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

Our algorithm then operates to identify the backup path from node q_i to any other 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, $a\in\{1,2,...,\sqrt{n}\}-\{x_i\}$, and $b\in\{1,2,...,\sqrt{n}\}-\{y_i\}$.

This computation can be performed by the common rendezvous node at position (x_j,y_i) since it is aware of the link state tables of all nodes in its partial sampling set; hence, it 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.2(a). 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, this 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.2(a), 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 $a\in\{1,2,...,\sqrt{n}\}-\{y_i\}$, respectively. Otherwise, q_a and q_b are in positions (a,y_i) and (a,y_j) for $a\in\{1,2,...,\sqrt{n}\}-\{x_i\}$, respectively. Fig.2(b) illustrates an example of all alternative paths between node 1 and node 4. In this way, every node q_i can find the best one among such $2\sqrt{n}-3$ alternative paths to node q_j according to its local information.



(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.

This computation can be locally done since node q_i knows the link state tables of all nodes in its partial sampling set $S(q_i)$; hence, 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.2(b). 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 that of $(3,4)$ from nodes 3 and 4.

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

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

As a summary, our efficient strategy for every node to send its partial probing results to all nodes in its partial sampling set provides enough information to identify an acceptable backup path for every node pair in the network. This is ensured by the two round operations at every node q_i, as shown in Algorithm 1. 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 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 every node pair in the network.

Theorem 1: This algorithm finds the backup path for every node pair in the network at the cost of every 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,

Algorithm 1 Finding backup paths at every node q

FirstRound()

1: **for** Any nodes q_i and q_j in the partial sampling set $S(q)$ **do**
2: Assume nodes q_i and q_j are in positions (x_i,y_i) and (x_j,y_j), respectively. It is clear that the common rendezvous node q_a must be in position (x_i,y_j) or (x_j,y_i).
3: Let *paths* denote the set of alternative paths for the direct path between q_i and q_j and be empty now.
4: **if** The two nodes are not in the same row or column. **then**
5: **for** a=1 to $\sqrt{n}$ **do**
6: Let $path_1$ be an alternative path from q_i to q_j with relay nodes in positions (a,y_i) and (a,y_j) in order.
7: Let $path_2$ be an alternative path from q_i to q_j with relay nodes in positions (x_i,a) and (x_j,a) in order.
8: Add $path_1$ and $path_2$ into the set *paths*.
9: Select the path with the lowest total latency from *paths* and notify the result to nodes q_i and q_j.

SecondRound()

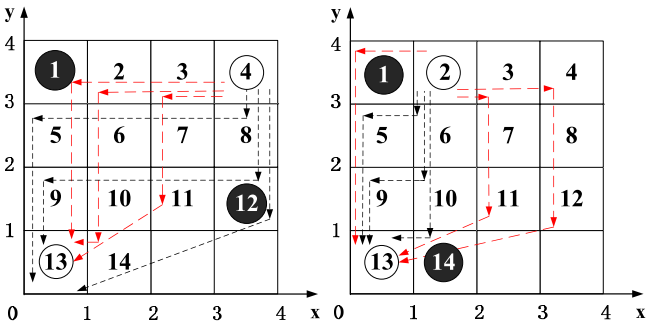
1: Let node q_i denote the current node q in position (x_i,y_i)
2: **for** Any node $q_j \in S(q_i)$ in position (x_j,y_j) **do**
3: Let *paths* denote the set of alternative paths for the direct path from q_i and q_j and be empty now.
4: **if** Nodes q_i and q_j are in the same row **then**
5: **for** $a = 1$ to $\sqrt{n}$ but $a \notin \{x_i,x_j\}$ **do**
6: Add *path* into *paths*, which is an alternative path from q_i to q_j with a relay node in position (a,y_i).
7: **for** $a = 1$ to $\sqrt{n}$ but $a \notin \{y_i\}$ **do**
8: Add *path* into *paths*, which is another path from q_i to q_j with two relay nodes in position (x_i,a) and (x_j,a).
9: **if** Nodes q_i and q_j are in the same column **then**
10: **for** $a = 1$ to $\sqrt{n}$ but $a \notin \{y_i,y_j\}$ **do**
11: Add *path* into *paths*, which is an alternative path from q_i to q_j with a relay node in position (x_i,a).
12: **for** $a = 1$ to $\sqrt{n}$ but $a \notin \{x_i\}$ **do**
13: Add *path* into *paths*, which is another path from q_i to q_j with two relay nodes in position (a,y_i) and (a,y_j).
14: Select the path with the lowest total latency from *paths* and notify the result to nodes q_i and q_j.

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 two bytes for latency, one byte for liveness and loss, and two bytes for every node ID. Thus, the link state table of every node is of size $10(\sqrt{n}-1)$ bytes. 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 two bytes for the ID of the destination node and four bytes for, at most, two relay nodes. Thus, every node q_i generates network traffic of $12(\sqrt{n}-1)^2$ bytes 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 every node pair in the network. □

3.4 Extension to networks with any number of nodes

The aforementioned construction strategy as well as the distributed implementation of the partial sampling scheme assumes that the number of nodes in the network is a perfect square, $\sqrt{n}\times\sqrt{n}$, such that the partial sampling set of every node is of size $2\sqrt{n}-2$. This assumption is usually not true in reality, resulting in empty spaces in the grid. We will show that our methodologies can be easily extended to networks with any number of nodes.



(a) From node 4 to node 13. (b) From node 2 to node 13.

Fig. 3. An illustrative example of the alternative paths for the direct path when the network size is not a perfect square.

Given a network with n nodes, we instead form a grid of c rows and $\lceil\sqrt{n}\rceil$ columns, where $c=\lceil\sqrt{n}\rceil$ or $\lfloor\sqrt{n}\rfloor$. If the last row contains only $d<\lceil\sqrt{n}\rceil$ nodes, the partial sampling set of any node in the last row of the grid may not have $\alpha=2\lceil\sqrt{n}\rceil-2$ elements. In this case, the first two conditions of Definition 1 fail to be satisfied for those node pairs, which involve at least one node in the last row of the grid. Consequently, there are only $c+d-2$ alternative paths between every node pair with one node in the last row. Furthermore, only one common rendezvous server exists for every node pair with one node in the last row while another node is in columns $d+1$ and beyond. Thus, the failure of the single rendezvous server in common will impair the computation and recommendation of backup paths for those node pairs.

To tackle this challenging issue, we revise the realization of the partial sampling set of every node q_i, for $1\le i\le d$, in the last row of the grid as follows. For any node q_i in position $(1,y_i)$, its partial sampling set $S(q_i)$ consists of $c-1$ other nodes in column y_i, $d-1$ other nodes in the last row, and other $\lceil\sqrt{n}\rceil-d$ nodes at positions $(2,d+1)$ to $(2,\lceil\sqrt{n}\rceil)$. In this way, $S(q_i)$ contains $c+\lceil\sqrt{n}\rceil-2$ elements, and the first two conditions of Definition 1 hold now. So far, every node q_i can measure its link state to all nodes in $S(q_i)$ and deliver its measured results to all nodes in $S(q_i)$.

With this method of construction, any two nodes have at least two common rendezvous nodes and $c+\lceil\sqrt{n}\rceil-2$ alternative paths for the direct path, no matter where the two nodes are located in the grid. Fig.3 plots an illustrative example of the alternative paths for the direct path when the network size is not a perfect square. The two nodes, 13 and 14, in the last row probe the node sets $\{1,5,9,14,11,12\}$ and $\{2,6,10,13,11,12\}$, respectively. For a node pair, 4 and 13, one of their common rendezvous nodes 16 does not appear and is replaced by node 12 according to the new construction scheme. So far, every common rendezvous node can select the one path with the lowest latency from 6 alternative paths for the node pair 4 and 13. For an alternative path $(4,3,11,13)$, the link states of $(4,3)$, $(3,11)$, and $(11,13)$ have been reported to

node 12 by node 4, node 11, and node 13, respectively. At the same time, the link states of $(4,3)$ and $(3,11)$ are reported to node 1 by node 3, while that of $(11,13)$ is reported by node 3 to node 1.

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 every 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 every node pair the best backup path among about $2\sqrt{n}-2$ alternative ones. The performance of the backup path for every 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 every 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. The two limits 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 every 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 more additional alternative paths for every node pair.

We present Definition 2 as an efficient construction method for the *enhanced partial sampling* based on the grid of size $\sqrt{n}\times\sqrt{n}$, which is formed by using the method mentioned in Section 3. Fig.4 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:

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

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

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.

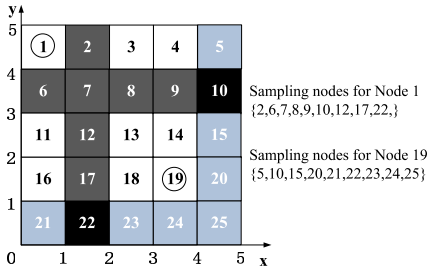


Fig. 4. An illustrative example of the enhanced partial sampling scheme.

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. This paper requires homogeneous implementations of the enhanced partial sampling sets for all nodes. Without loss of generality, we assume $k=1$ and 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 2 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 2 nodes. This generates $\sqrt{n}$ or 2 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 bringing $\sqrt{n}$ or 2 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 selecting using enhanced partial sampling

Although the enhanced partial sampling can potentially provide some more alternative paths for a large majority of the node pairs in the network, this benefit can be implemented only through carefully designed approaches that involve three basic stages. In the first stage, every node q_i measures its links 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 observed by every node to find the backup path for every node pair, the following two stages conducted at every node are essential. They are the dissemination of measuring results and the decision-making about the backup

path. Actually, they have the same functionalities as the last two stages in our original approach in Section 3.3, but 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 every node pair, q_i and q_j, there exist 2 or $\sqrt{n}$ common rendezvous nodes, each of which can identify the 2 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 since 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 that from node 6 among $Es(1)$, in Fig.4, if every node only sends its link state table to the nodes in its enhanced partial sampling set. Thus, this backup path selecting method meets the same limit that is also faced by our prior approach based on partial sampling: there is an insufficient number of alternative paths for every 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 every node pair, q_i and q_j, derived from the second condition of the enhanced partial sampling. We show the details from the following three aspects. We can see that the selected path for the direct path from q_i to q_j is different from that from 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, 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 common rendezvous node, while another node, in position (x_j, y_i^+), is a redundant one. For example, as shown in Fig.5, 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 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 the position $(x_a, y_a = y_i^+)$, and node q_b is in positions (x_a^+, y_j^+) and (x_j^+, y_a^+). Here, $x_a \in \{1, 2, ..., \sqrt{n}\} - \{x_j, x_j^+\}$.
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_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 with the node in position (x_j^+, y_i^+) being 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.5, the right column demonstrates all alternative paths derived by the preferred common rendezvous node 22.

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. 5. 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. 6. 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.

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, $y_a \in \{1, 2, ..., \sqrt{n}\} - \{y_j, y_j^+\}$.
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 with the node in position (x_i^+, y_j^+) being a relay node.

In the case that q_i and q_j are in the same row, they are aware of each other's link state table, 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 $Es(q_i)$. It is clear that another node 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 $y_a \in \{1, 2, ..., \sqrt{n}\}$. 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 4 are demonstrated by the left column in Fig.6, where nodes 7 and 8 are two rendezvous nodes in common.

In the case that 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 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 $x_a \in \{1, 2, ..., \sqrt{n}\}$. 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 by the right column in Fig.6, 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: In the case that 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.6. 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 every 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 $Es(q_i)$, hence resulting in $4\sqrt{n}-5$ messages for a total size of $40n-70\sqrt{n}+25$ bytes. This gives sufficient information to identify a good backup path for every node pair in the network through three round operations at every node q_i.

In the first round, node q_i discovers the best backup path for every 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 $2\sqrt{n}-2$ 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 two bytes for the ID of the destination node and four bytes for, at most, two relay nodes. Thus, every q_i generates network traffic of $12(\sqrt{n}-1)^2$ bytes 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^-), hence resulting in network traffic of $12(\sqrt{n}-1)$ bytes. 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 $8\sqrt{n}-7$ total messages and $52n-66\sqrt{n}+17$ bytes so as to identify one good backup path for every node pair in the network. □

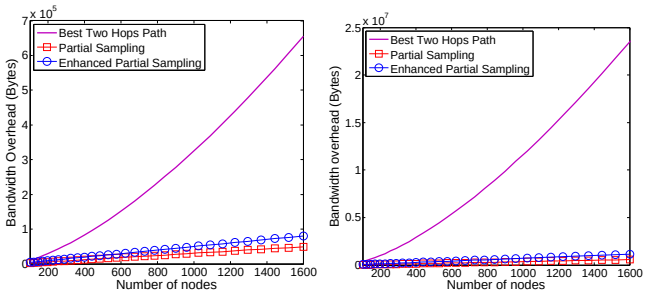
4.3 Rotational partial sampling

The enhanced partial sampling significantly increases the number of alternative paths for every node pair in the network. Every node, however, always measures the same set of nodes, hence 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.

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 the increasing of k (the round of sampling). The value of k is reset to 1 once it exceeds $\sqrt{n}$ since $Es(q_i, k_1) = Es(q_i, k_2)$ 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.

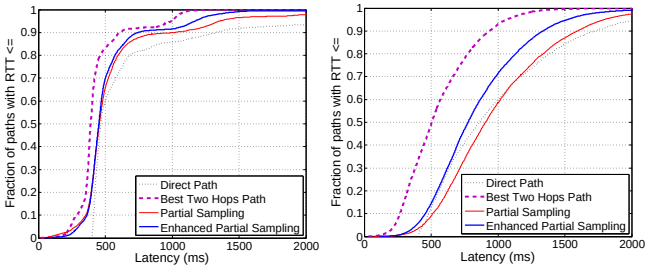
At the same time, every node q_i achieves the entire view about its link states to the rest of the nodes in the network after $\sqrt{n}$ rounds. However, only about $1/\sqrt{n}$ of the observed view is refreshed while other parts become historical records. Additionally, the two common rendezvous nodes in positions (x_i, y_j) and (x_j, y_i) are aware of the link status from any node q_i in the position (x_i, y_i) to all other nodes and from any node q_j in the position (x_j, y_j) to the rest nodes in the network. Therefore, the path selecting approach, based on rotational partial sampling, can be improved to exhibit better performance if some part of the historical measuring results of every node are utilized. As we will show, it can increase the probability that the resultant backup path for any node pair does not experience worse than the direct path to about 85%, even if it only uses the measuring results of every node during the current and last rounds. This probability is sufficiently high and can be further increased if more historical measuring results of every node are used. Additionally, for any node pair, another backup path will be discovered in the next round, and the backup paths exhibit lower performance than the default path in two continuous rounds with very low probability, only 2.5%.

On the other hand, the historical measuring results by every node q_i can be utilized to derive some statistical models [20], [21], 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



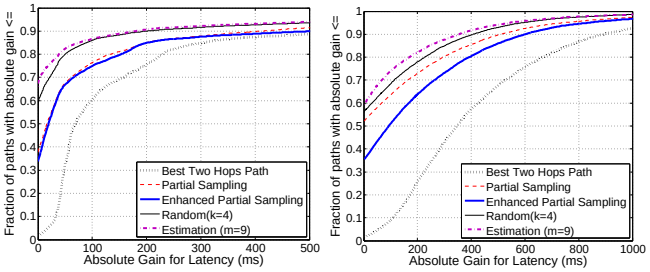
(a) Measure other nodes using the ping. (b) Measure other nodes using the traceroute.

Fig. 7. Comparison of the average amount of network traffic incurred by per-node in each round.



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

Fig. 8. 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. 9. The absolute gain of latency on average, due to the introduction of the backup path for every node pair.

statistical models are complementary to our approaches since the delays of partial or whole unmeasured paths of every node can be predicted at a given level of accuracy. This works well, especially for those paths that are not measured at the current round but are measured recently, especially in the latest round. 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 every node pair. We leave this research issue as one for our future work.

4.4 Extension to networks with any number of nodes

The approach based on the enhanced partial sampling meets the same problem that was mentioned in Section 3.4. That is, the number of nodes in the network is usually not a perfect square, $\sqrt{n}\times\sqrt{n}$, resulting in empty spaces in the grid. We will show that our approach can be easily extended to networks with any number of nodes after minimal modifications.

Given a network with n nodes, we instead form a grid of c rows and $\lceil\sqrt{n}\rceil$ columns, where $c=\lceil\sqrt{n}\rceil$ or $\lfloor\sqrt{n}\rfloor$. If the last row contains less than $\lceil\sqrt{n}\rceil$ nodes, this will impair the implementation of the approach that is based on enhanced partial sampling. One efficient way would be for every empty space in the grid to find an existing node in other space as its virtual node. Although there may exist many different ways to realize this idea, without loss of generality, we assume that the node on the top of every empty space acts as its virtual node. Thus, the path selecting approach we presented in this section can be utilized directly, even if the number of rows is less than that of columns.

5 Evaluation

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

5.1 Description of datasets

A. PlanetLab Trace. This trace shows the maximum, average, and minimum latencies (over 10 pings in a 15 minute interval) between all node pairs on PlanetLab [22] from January 2004 to June 2005 (data from [23]). 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 [24] service publishes the traceroute results from 200 source nodes to 140,000 destination nodes every day. All source nodes are PlanetLab nodes, which also appear in the set of destination nodes. Actually, the set of destination nodes contains large number of non-PlanetLab nodes besides the source nodes (data from [25]). After collecting the iPlane trace from April 1, 2011 to May 30, 2011, we extract an archive of traceroute between every node pair on 169 PlanetLab nodes for our evaluations. Note that the published results in iPlane [24] only contain the all-pairs traceroute results among at most 200 PlanetLab nodes.

5.2 Overhead: bandwidth consumption

We first evaluate the per-node bandwidth consumption of our backup path selecting approaches, which are based on partial sampling and enhanced partial sampling schemes, and the approach that finds the best two-hops path[2] in [6], which is the best one before our proposals. It is worth noticing 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 and hence is not evaluated again. We perform this evaluation by using in-system emulations, under both the first case, where every node probes other nodes using the ping operation, and the second case using the traceroute operation. The experimental results are plotted in Fig.7.

2. Actually, it means that this path traverse two relay nodes.

Theorem 3: The approach to finding the best backup path with two overlay hops in [6] makes every node incur $n+4\sqrt{n}-3$ messages and $10n\sqrt{n}+10n-34\sqrt{n}+14$ bytes.

Proof: First of all, this approach requires every node q_i to measure its link states to all other nodes in the network. This generates $n-1$ total messages for a total of $8n-8$ bytes. As a result, a link state table of size $5(n-1)$ bytes for every node is formed. Furthermore, node q_i sends its link state table to all nodes in $S(q_i)$, and this results in $2(\sqrt{n}-1)$ messages for a total size of $10(n-1)(\sqrt{n}-1)$ bytes. Finally, node q_i sends routing recommendations to all nodes in $S(q_i)$, hence causing $2(\sqrt{n}-1)$ messages for a total size of $12(\sqrt{n}-1)^2$ bytes. Thus, every node incurs $n-1+4(\sqrt{n}-1)=n+4\sqrt{n}-3$ messages and $10n\sqrt{n}+10n-34\sqrt{n}+14$ bytes. Theorem 3 holds. □

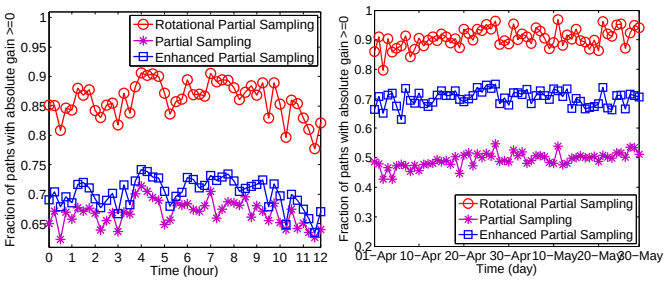
We can see that our approaches indeed dramatically reduce the per-node bandwidth consumption compared to the prior approach in [6], irrespective of the network size. Additionally, the experimental results also confirm the theoretical results proved by Theorems 1, 2, and 3. This demonstrates that our approaches scale the network as expected. The theoretical per-node bandwidth consumption of the three approaches can be inferred in the similar way for the second case; hence, the details are omitted. Note that the per-node bandwidth consumption of our approach using the enhanced partial sampling is a little bit more than that uses the partial sampling, as was expected.

5.3 Effectiveness

We then perform a measurement study on the latencies of the direct Internet path and the indirect backup paths from four approaches for every node pair. They are the best two-hops path selecting approach in [6], the first two approaches presented in this paper, the random selection approach with k=4 in [7], and the enhanced earliest-divergence approach with m=9 in [13] (called estimation approach here). Fig. 8 plots the CDF of path latency for different settings.

We first extract 9241 pairs of communicating nodes from the PlanetLab trace whose end-to-end latencies along the direct Internet paths are larger than 400ms. Fig.8(a) shows the improvement in latency given by the best two-hops path and the backup path from our approaches for those 9241 direct Internet paths. Fig.8(b) shows that for 14558 direct Internet paths whose point-to-point latencies are larger than 400ms in the iPlane trace. We can see that our two approaches introduce 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.8, and our approaches outperform them, as shown in Fig.9.

With these measured results, we compare our approaches with others from the aspect of the absolute gain. Here, the latency on the direct path minus the latency of the backup



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

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

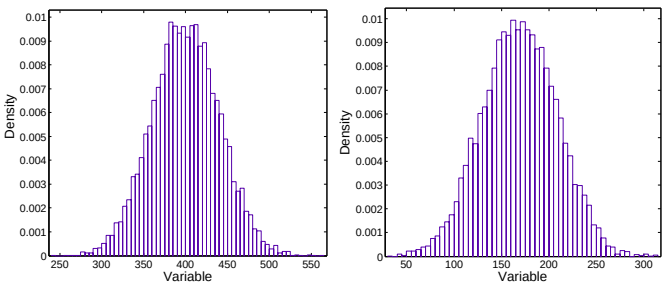
Fig. 10. A comparison between approaches in their ability to pick backup paths which have the same or an even better performance than the direct paths.

path, recommended by different approaches, is defined as the absolute gain. Fig.9 plots the CDF of absolute gain for the two different settings. We can see that the best two-hops path approach almost always finds a backup path exhibiting lower latency than the direct path for every 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-hops alternative paths for every 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%, in comparison to the latency of the direct path for every 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-hops path approach. The root reason is that every node just measures at most $2\sqrt{n}$ other nodes, and hence only, at most $6\sqrt{n}$, alternative paths can be identified for the direct path between every node pair.

To improve the performance of each recommended backup path, we further propose the path selecting approach based on *rotational partial sampling*. As shown in Fig.9, for every node pair, the introduction of rotational partial sampling would increase the probability that the backup path exhibits a similar to or even better performance than the direct path to about 85%, even if it only uses the measuring results of every node during the current and last rounds. For many network applications, this is sufficiently high such that the increased scalability of networks outweighs this drawback. Additionally, it is not necessary to absolutely identify an outstanding backup path for every node pair in each round since 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 every node pair exhibits lower performance than the default path in two continuous rounds is very low.

To validate our new approach in wide scenarios, we perform the evaluation on the two traces over a relatively long period. Fig.10 shows that our new approach achieves a stable improvement in the delay of selected backup paths under the two evaluation settings, even if it only uses the measuring results of every node during the current and last rounds.



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

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

Fig. 11. The distribution of a variable that denotes the number of relay load.

6 Conclusion

Recent efforts have shown that 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 every node pair, with a probability of about 85%. For many network applications, this is sufficiently high such that the improved scalability of networks outweighs this drawback. The approach that we presented offer an exciting step in scaling full-mesh overlays and can promote their usages in wider classes of applications.

Following the work in this paper, 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 face of node and link failures. Secondly, we will redesign our approaches for finding a backup path that is not heavily correlated with the direct path.

References

[1] H. Li, L. Mason, and M. Rabbat, "Distributed adaptive diverse routing for voice-over-ip in service overlay networks," *IEEE Transactions on Network and Service Management*, vol. 6, no. 3, pp. 175–189, 2009.

[2] F. Wang and L. Gao, "A backup route aware routing protocol-fast recovery from transient routing failures," in *Proc. 27th INFOCOM*, Phoenix, AZ, USA, Apr. 2008, pp. 2333–2341.

[3] ——, "Path diversity aware interdomain routing," in *Proc. 28th INFOCOM*, Rio de Janeiro, Brazil, Apr. 2009, pp. 307–315.

[4] K.-W. Kwong, L. Gao, R. Guerin, and Z.-L. Zhang, "On the feasibility and efficacy of protection routing in ip networks," in *Proc. 29th INFOCOM*, San Diego, California, USA, Mar. 2010.

[5] 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.

[6] D. Sontag, Y. Zhang, A. Phanishayee, D. G. Andersen, and D. Karger, "Scaling all-pairs overlay routing," in *Proc. ACM CoNEXT*, Rome, Italy, Sep. 2009.

[7] 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," in *Proc. 6th OSDI*, San Francisco, California, USA, Dec. 2004, pp. 183–198.

[8] D. Guo, J. Wu, Y. Liu, H. Jin, H. Chen, and T. Chen, "Quasi-kautz digraphs for peer-to-peer networks," *IEEE Transactions on Parallel Distributed Systems*, vol. 22, no. 6, pp. 1042–1055, 2011.

[9] A. Nakao, L. L. Peterson, and A. C. Bavier, "Scalable routing overlay networks," *Operating Systems Review*, vol. 40, no. 1, pp. 49–61, 2006.

[10] R. S. Kazemzadeh and H.-A. Jacobsen, "Adaptive multi-path publication forwarding in the publiy distributed publish/subscribe systems," University of Toronto, Canada, Tech. Rep., Nov. 2011.

[11] D. G. Andersen, H. Balakrishnan, M. F. Kaashoek, and R. Morris, "Resilient overlay networks," in *Proc. 18th ACM Symposium on Operating Systems Principles (SOSP)*, anff, Canada, Oct. 2001, pp. 131–145.

[12] R. Friedman, G. Kliot, and C. Avin, "Probabilistic quorum systems in wireless ad hoc networks," *ACM Transactions on Computer Systems*, vol. 28, no. 3, pp. 184–206, 2010.

[13] T. Fei, S. Tao, L. Gao, and R. Guérin, "How to select a good alternate path in large peer-to-peer systems?" in *Proc. 25th INFOCOM*, Barcelona, Catalunya, Spain, Apr. 2006.

[14] G. Wang, B. Zhang, and T. S. E. Ng, "Towards network triangle inequality violation aware distributed systems," in *Proc. 7th ACM SIGCOMM Conference on Internet Measurement (IMC)*, San Diego, California, USA, Oct. 2007, pp. 175–188.

[15] C. Lumezanu, R. Baden, N. Spring, and B. Bhattacharjee, "Triangle inequality variations in the internet," in *Proc. of the 9th ACM SIGCOMM Conference on Internet Measurement (IMC)*, Chicago, Illinois, USA, Nov. 2009, pp. 177–183.

[16] 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," in *Proc. 29th IEEE INFOCOM*, San Diego, CA, USA, Mar. 2010, pp. 291–295.

[17] F. Cantin, B. Gueye, D. Kaafar, and G. Leduc, "Overlay routing using coordinate systems," in *Proc. 2008 ACM Conference on Emerging Network Experiment and Technology (CoNEXT)*, Madrid, Spain, 2008.

[18] W. Cui, I. Stoica, and R. H. Katz, "Backup path allocation based on a correlated link failure probability model in overlay networks," in *Proc. 10th IEEE International Conference on Network Protocols (ICNP)*, Paris, France, Nov. 2002, p. 236.

[19] W. W. Terpstra, J. Kangasharju, C. Leng, and A. P. Buchmann, "Bubblestorm: Resilient, probabilistic, and exhaustive peer-to-peer search," in *Proc. SIGCOMM*, Kyoto, Japan, Aug. 2007, pp. 49–60.

[20] B. D. Abrahao and R. D. Kleinberg, "On the internet delay space dimensionality," in *Proc. the 8th ACM SIGCOMM Conference on Internet Measurement (IMC)*, Vouliagmeni, Greece, Oct. 2008, pp. 157–168.

[21] 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.

[22] M. Olbrich, F. Nadolni, F. Idzikowski, and H. Woesner, "Measurements of path characteristics in planetlab," Technical University Berlin, Berlin, Tech. Rep. TKN Technical Report TKN-09-005, Jul. 2009.

[23] All-pairs-pings. [Online]. Available: http://pdos.csail.mit.edu/štrib/

[24] H. Madhyastha, E. Katz-Bassett, T. Anderson, A. Krishnamurthy, and A. Venkataramani, "iPlane Nano: Path prediction for peer-to-peer applications," in *Proc. 6th NSDI*, Boston, MA, USA, 2009, pp. 137–152.

[25] Datasets of iPlane. [Online]. Available: http://iplane.cs.washington.edu