Robust Backup Path Selection in Overlay Routing with Bloom Filters

Xiaolei Zhou, Deke Guo, Tao Chen and Xueshan Luo
Science and Technology on Information Systems Engineering Laboratory
National University of Defense Technology
Changsha, 410073-China
[e-mail: zhouxiaolei@nudt.edu.cn]
*Corresponding author: Xiaolei Zhou

Received April 3, 2013; revised June 17, 2013; revised July 22, 2013; accepted August 16, 2013; published August 30, 2013

Abstract

Routing overlay offers an ideal methodology to improve the end-to-end communication performance by deriving a backup path for any node pair. This paper focuses on a challenging issue of selecting a proper backup path to bypass the failures on the default path with high probability for any node pair. For existing backup path selection approaches, our trace-driven evaluation results demonstrate that the backup and default paths for any node pair overlap with high probability and hence usually fail simultaneously. Consequently, such approaches fail to derive a robust backup path that can take over in the presence of failure on the default path. In this paper, we propose a three-phase RBPS approach to identify a proper and robust backup path. It utilizes the traceroute probing approach to obtain the fine-grained topology information, and systematically employs the grid quorum system and the Bloom filter to reduce the resulting communication overhead. Two criteria, delay and fault-tolerant ability on average, of the backup path are proposed to evaluate the performance of our RBPS approach. Extensive trace-driven evaluations show that the fault-tolerant ability of the backup path can be improved by about 60%, while the delay gain ratio concentrated at 14% after replacing existing approaches with ours. Consequently, our approach can derive a more robust and available backup path for any node pair than existing approaches. This is more important than finding a backup path with the lowest delay compared to the default path for any node pair.

Keywords: Backup path selection; Overlay Network; Overlay Routing; Bloom filters

A preliminary version of this paper appeared in proceedings of IEEE HPCC 2012, June 25-27, Liverpoor, England. This version includes an improvement of the scalability with the Bloom filter. This research is supported in part by the NSF China under Grant No. 61170284, 61202487, 71071160, and National Basic Research Program (973 program) under Grant No. 2014CB347800.

http://dx.doi.org/10.3837/tiis.2013.08.009

1. Introduction

Path diversity [1] is an effective way to improve the end-to-end performance of the modern Internet applications, especially for these delay-sensitive applications, such as Voice-over-IP (VoIP) [2] and online multimedia sharing. Utilizing alternative paths adaptively can avoid the link failures frequently occurring on the communication path. When the default path between any communicating node pair fails, a backup path selected in advance will take over to forward the packets. The current TCP/IP protocol, however, does not naturally support multipath routing. Therefore, many approaches about finding and utilizing the backup path have been done on the overlay networks to realize the multipath routing scheme. Such approaches fall into one of two categories according to the metric used to selecting the backup path: (a) Delay-based approaches [3-5] that rely on the delay of different backup paths, and (b) Topology-based approaches that utilize the topology information of the network, i.e., BGP (Border Gateway Protocol) information [6] or IP level path information [9].

The Delay-based approaches select an alternative path with the minimum delay, but seldom concern about the correlation between the alternative path and the default path, which may make the selected backup path fail together with the default path. It is crucial to consider about the path correlation, as the more the backup path overlaps with the default path, the more possibly that the backup path fails simultaneously with the default path. This paper conducts a long-term observation based on two real dataset of the PlanetLab. The results indicate that the backup path selected by the delay-based approach overlap with the default path with a probability of over 75%, and the backup path fail simultaneously with the default path with a probability of 31.2% on average.

In contrast, the topology-based approaches focus on selecting a backup path with as little correlation with the default one as possible. However, the AS level path inference based on the BGP information is uncertainty and may miss those potential better paths. On the other hand, the IP level path inference based on the *traceroute* probing [9] incurs large communication overhead, and thus is lack of feasibility in practice.

In this paper, we focus on how to select a more proper and robust backup path to improve the end-to-end communication performance. The desired backup path should have the least overlap with the default path, and exhibits a relatively lower delay. To find such a path, the *traceroute*, a common light-load probing protocol, can provide the fine-grained topology information, but does bring extensive communication overhead. Two challenging issues arising here are as follows. Firstly, how to identify a more proper backup path according to the overlap between the default and backup path? Secondly, how to reduce the resulting communication overhead in order to satisfy the scalability requirement of the modern network applications? To tackle such issues, we propose a robust backup path selection approach, named RBPS.

Note that we prefer the alternative path to be a one-hop path, rather than a multi-hop path. The one-hop path forwards the message through an additional relay node. As shown in **Fig. 1**, the direct path from node s to node d is named as the default path, while the path from node s through node i and finally to node d is a one-hop alternative path. The delay of a multi-hop path is usually larger than that of a one-hop path between the same node pair [4]. Moreover, we believe that a proper one-hop path should be adequate to detour around the failed physical links on the related default path with high probability. In the remainder of this paper, the

one-hop path is named as a candidate path, while the selected candidate path is named as a backup path.

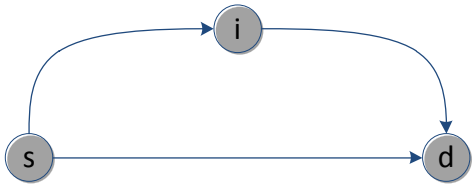


Fig. 1. An illustrative example of one-hop backup path.

Our RBPS approach utilizes the delay and path overlap as two criteria, to select the backup path for any node pair in a heuristic manner. The effectiveness of a backup path requires as little path overlap with the related default path as possible, while the delay-sensitive applications prefer the backup with low delay. A single alternatives path, however, seldom minimizes the two criteria at the same time. Note that, it is possible that the selected backup path fails with the default path together. Therefore, the availability is primary for a backup path. Otherwise, no matter how low the delay of the selected backup path is, it does not work. This paper proposes a heuristic selection approach that selects an alternative path with the minimum delay among those paths whose overlap ratio, compared with the default path, are below a given upper bound. The existence of such a one-hop path will be demonstrated via a long-term trace-driven observation.

Consider that the *traceroute* protocol incurs non-trivial communication overhead due to frequently probe and distribute the link state from each node to all of other nodes for selecting a backup path for any node pair. Our RBPS approach, therefore, employs the grid quorum [5] as a fundamental service to determine the destination nodes to distribute the link-state tables from each node. Owing to the grid quorum, the size of the destination set is reduced to $2(\sqrt{N}-1)$, where N denotes the network size.

To further reduce the per-node communication overhead, we utilize Bloom filters to compress the link-state table at each node before disseminating to other nodes. We further study how to accurately estimate the path overlap between two paths that have been presented as Bloom filters. The parameter settings of such Bloom filters are then optimized via extensive experiments. To the best of our knowledge, we are the first to employ the Bloom filter to estimate the actual path overlap length between two paths. The path overlap estimation approach can be easily migrated to the other applications.

On the foundation of the above ideas, we develop the RBPS approach to improve the end-to-end communication performance for the delay-sensitive network applications. The RBPS is robust, scalable and feasible. It identifies a more proper backup path that can detour around the failures on the default path, while exhibits a near minimum delay among all the candidates. The per-node communication overhead for identifying such a path is bounded by $O\ (n^{1.5})$, which is desirable for improving the fault-tolerate ability of the backup path.

Extensive trace-driven evaluations are conducted based on the iPlane and all-pairs-ping dataset. Compared with the best delay-based approach, SAP [5], the simultaneous failure probability is reduced by about 60 percent, while the delay of the selected backup path increases slightly. The delay gain ratio of the selected backup path is concentrated at 14 %. Compared with the delay-based approach, RON [3], the communication overhead is reduced by over 80 percent. Furthermore, when the network size reaches 1,000, the RBPS incurs about 1MB extra per-node communication overhead, even compared with the current least overhead approach. For the path overlap estimation based on the Bloom filter, the relative estimation error is below 5%, while the false positive rate of the Bloom filter is below 0.01.

The rest of this paper is organized as follows. Section 2 summarizes the most related work of this paper. Section 3 introduces the preliminaries of the grid quorum system and the Bloom filter. Section 4 reveals the long-term observations on the simultaneous failure problem in the delay-based approach, and analyzes the root cause. Section 5 proposes the RBPS approach, including an overview of our approach, the heuristic backup path selection and the representation of each link-state table via the Bloom filter. Section 6 studies the simultaneous failure probability, delay gain, overlap and the communication overhead of our approaches via extensive trace-driven emulations. We conclude this work in Section 7.

2. Related Work

Identifying a backup path with better performance in the overlay routing has attracted tremendous interests in the past two decades. The conventional backup path selection approach, employed in RON [3] and Tapestry [8], makes each node send *ping* message to all the other nodes every short time interval, e.g., 15 minutes. Once the probing results return, a link-state table is maintained locally and distributed to all the other nodes. In that case, each node in the network will receive the link-state tables from all the other nodes and hence can identify the best candidate path for an arbitrary node pair. Such an approach generates a probing and communicating overhead of $O(N^2)$ per-node, where N denotes the network size, which limits its scalability. Besides, it does not consider about the path overlap between the default and backup paths, and thus suffers from the simultaneous failure problem.

Gummadi et al. [4] propose an approach based on the correlated failure probability model, which achieves an acceptable accuracy when the failure probability is a prior knowledge resulting from long-term observations. It is clear that such an approach requires a long term measurement for determining the correlated failure probability. However, such an assumption is not feasible in large-scale distributed systems.

Zhang et al. [9] employ the IP address information as well as the delay to help selecting the backup path with the maximum disjointness. It is regarded as the first attempt to utilize IP address sequence to estimate the overlap between the default and backup paths. However, this approach suffers from the extra-large communication overhead and thus lack of feasibility in practice.

Nakao et al. [10] take the path overlap into consideration, and employ the BGP (Border Gateway Protocol) information to discover a disjoint AS (Autonomous System) level alternative path. Similarly, Fei et al. [6] propose a heuristic approach, based on an intuitive earliest-divergence rule, to identify an AS level backup path. The basic idea is to find a backup path that diverge with the default one earliest, or detours farthermost from the default path. Those approaches are light load and achieve the better scalability. To some extent, the approaches based on the AS level path information have addressed the simultaneous failure problem, as well as the scalability limitation. They, however, are instinctively coarse-grained approaches, and will possibly miss the most proper candidate path.

Recently, Sontag et al. [5] propose the scaling all-pairs (SAP) overlay routing approach, which is considered as the current best approach. It employs the grid quorum system to distribute the link-state tables and hence largely reduces the per-node communication overhead to $O(N^{1.5})$. The SAP approach, however, selects the backup path merely based on the delay criterion; hence, it cannot properly address the simultaneous failure problem as well. In this paper, the SAP approach is taken as a baseline of our approaches.

In summary, neither the delay-based approaches nor the BGP information based approaches are adequate to yield an appropriate backup path, to detour around the failure on the default path. Compared with the above work, we propose a fine-grained approach to identify a disjoint backup path with a reasonable communication overhead.

3. Preliminary

3.1 Grid Quorum System

Quorum systems [11, 12] have been used in the field of distributed control and management, such as the mutual exclusion, data replication protocols, and secure access control. The grid quorum system is a specific quorum system, where each quorum is defined based on a grid. It is firstly introduced to the backup path selection in [5] to reduce the communication overhead. The definition of a grid quorum system is given in Definition 1. **Fig. 2** shows an illustrative example of a grid quorum system of 16 nodes.

Definition 1. (Grid quorum system) Suppose that the network size is a perfect square number, denoted as N, and all the N nodes are located in a $\sqrt{N} \times \sqrt{N}$ grid. For each node at (i, j), Quorum $Q_{i,j}$ consists of all the nodes in ith row and jth column except the node (i, j).

For convenience, this paper just discusses the grid quorum system that the network size is a perfect square number. The case that the network size is a non-perfect square number is similar. More information about can be found in [5].

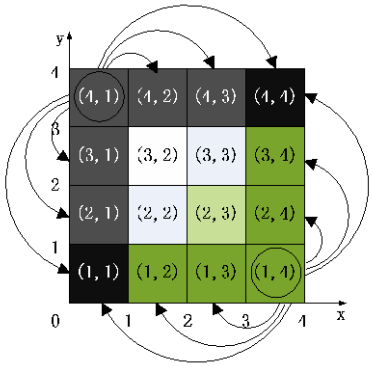


Fig. 2. An illustrative example of a grid quorum system of 16 nodes.

For any two different grid quorums $Q_{i,j}$ and $Q_{u,v}$, the elements in their intersection are named as the rendezvous nodes. Note that, there are at least two rendezvous nodes for any two grid quorums. Node (i, v) acts as the preferred rendezvous node, while the other one acts as a redundant. When each node exchanges the link-state tables with the grid quorum system, the per-node communication overhead can be reduced from $O(N^2)$ to $O(N^{1.5})$. For every node pair, the rendezvous node receives the link-state tables from both sides of the node pair, and thus is able to identify the best one-hop backup path among the $N - 2$ candidate paths. Actually, the configuration of the grid quorum system can be various. This paper just uses the configuration in [5] as the fundamental service of our proposal.

3.2. Bloom Filter

Bloom filter [13-15] is a compact data structure, which is widely used in the membership checking of a large dataset with a low memory overhead. The basic idea of Bloom filter is to employ a vector V of m bits to encode the membership of a set $X = \{x_1, x_2, \ldots x_n\}$. The

values of all the bits in vector V are initialized to be $'0'$. And then, k independent hash functions $h_1, h_2 \ldots h_k$ are taken, each of which ranges from $\{1 \ldots m\}$. For each element $x \in X$, the bits at positions $h_1(x), h_2(x) \ldots h_k(x)$ in vector V are set to be $'1'$. In this way, responding a membership query like "*whether* x' *is an element of set* X", just need to examine whether all the bits at positions $h_1(x'), h_2(x') \ldots h_k(x')$ in V are all set to be $'1'$. If not, we can definitely infer that the element x' does not belong to the set X. Otherwise, the element x' possibly belongs to the set X.

Note that x' may not belong to the set X even all bits at positions $h_1(x'), h_2(x') \ldots h_k(x')$ are set to be $'1'$. Such elements are defined as false positive elements. The root cause is that multiple elements in the set X may make the bits at $h_1(x'), h_2(x') \ldots h_k(x')$ in V set to be $'1'$. The false positive probability f_p can be calculated by Equation (1) in [14]. For many applications, it will be acceptable as long as the false positive probability remains sufficiently low. As pointed out in [14], when $k = (ln\ 2) \cdot ({}^{m}/_{n})$, the false positive probability f_p is minimized to $(0.5)^k = (0.6185)^{m/n}$.

$$f_p = (1 - (1 - \frac{1}{m})^{kn})^k \approx (1 - e^{\frac{kn}{m}})^k \quad (1)$$

B. Donnet et al. [16] firstly utilize the Bloom filter to compress the raw *traceroute* information for evaluating the path similarity. Each path is represented by a Bloom filter and the similarity between two different paths is measured by the *Bloom Distance*, which is defined as the ratio of the number of $'1'$ in the intersection of the two Bloom filters to the sum number of bits in the two Bloom filters. The *Bloom Distance*, however, is an abstract measurement of path similarity, which indicates how much the two paths are alike, rather than the actual length of the path overlap.

Compared with the *Bloom Distance*, the backup path selection requires the actual length of path overlap to identify a more disjoint path. On this basis, we propose a novel approach to estimate the actual overlap length between two paths encoded by the Bloom filters. Our approach can be used as a fine-grained path similarity measurement. Besides, the one-hop paths that represented by a Bloom filter cannot be obtained directly, and thus require operations to combine two independent paths. On this basis, the path overlap estimation approach is proposed in our RBPS approach.

4. Observations on the Simultaneous Failure Problem

A long-term observation is conducted to investigate the simultaneous failure problem with two real datasets on the PlanetLab testbed. In the experimental setting, the backup path is selected in advance by the SAP approach [5], a current best delay-based approach.

A. All-pair-ping dataset

The all-pairs-ping [17] dataset contains the end-to-end delay information of each node pair on the PlanetLab testbed. It makes every node send over 10 *ping* messages to the other nodes every 15 minutes, and records the maximum, average and minimum values of the end-to-end delay for every node pair within about 16 months. We exact a subset from April 1, 2005 to April 4, 2005, which contains about 440 PlanetLab nodes, to observe the actual simultaneous failure problem.

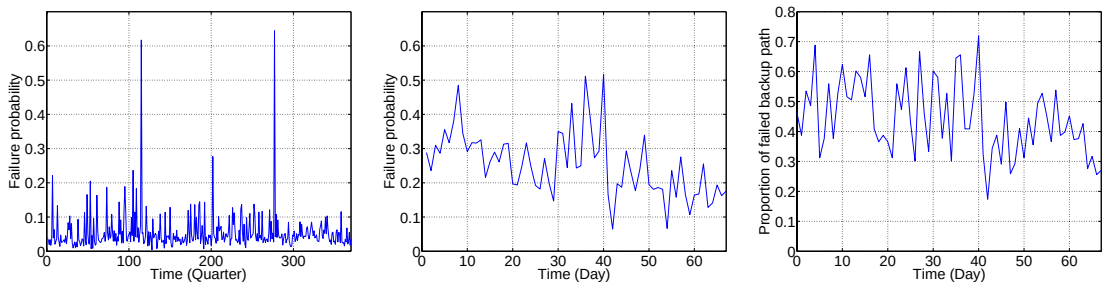
B. iPlane dataset

The iPlane dataset [18] is published by the iPlane [19] service. It contains the daily *traceroute* probing results from about 200 vantage nodes to 140,000 destination nodes on the edge of the Internet. All of the vantage nodes are also included in the destination nodes.

Therefore a dataset of all-pairs *traceroute* can be constructed. We collect the iPlane trace from March 1, 2011 to May 10, 2011, and extract such an archive of about 200 PlanetLab nodes for our observation.

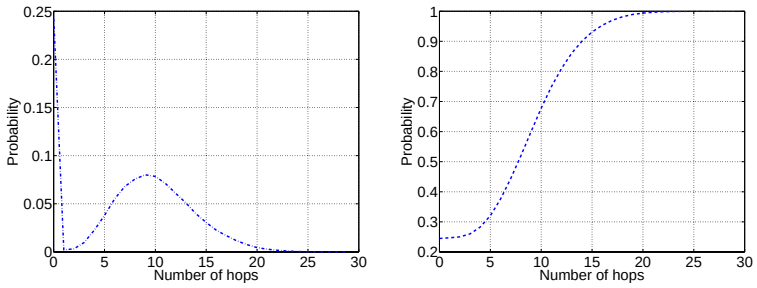
We first evaluate the simultaneous failure problem based on the two datasets. The backup path for each node pair is selected according to all of link-state tables at a previous moment. For the all-pairs-ping dataset, if the delay of a default path at moment t is less than a certain threshold (100,000ms in the experiment), the path will be considered as failed. If the delay of the backup path is also beyond that threshold, the backup path is treated as failed as well. For the iPlane dataset, when the returned IP sequence is NULL or the delay between two hops is beyond a certain threshold (1000ms in the experiment), the path is considered as failed. Actually, the above two conditions to determine whether the path is failed are coincident. When the *traceroute* returns NULL, the delay definitely reaches an unavailable value.

Fig. 3(a) and **Fig. 3(b)** plot the simultaneous failure probabilities. Due to the difference in the network states (topology and network traffic) at the time the traces are collected and sampling period, the two observations show different simultaneous failure probabilities. More preciously, it is 13.4% on average for the all-pairs-ping dataset, while 31.2% on average for the iPlane dataset. It shows that the backup paths selected merely based on the delay criterion, frequently fails with the default path at the same time. More seriously, such a probability may even approach more than 60% and 50% in the all-pairs-ping and iPlane datasets, respectively. Such a failure probability will make the backup path fail to take over the failed default path. Obviously, the backup path selection approaches merely based on the delay criterion are not adequate to identify an appropriate backup path.



(a) All-pairs-ping observation. (b) iPlane observation. (c) Propotion of the failed paths owing to the path overlap.

Fig. 3. Simultaneous failure probabilities of the selected backup path for the delay-based approach.



(a) PDF (b) CDF

Fig. 4. PDF and CDF of the overlap between the default and backup paths for SAP approach.

In order to find out the root cause of the simultaneous failure problem, the path overlap between the default and backup paths is investigated based on the iPlane dataset. **Fig. 4(a)** plots the Probability Density Function (PDF) of the overlap between the backup and default path. The overlap concentrates at 9 hops with a probability of 8.01%. **Fig. 4(b)** plots the Cumulative Distribution Function (CDF). Surprisingly, the backup path may overlap with the default path with a probability of 75.5%, while the maximum overlap between the default and backup paths even achieves 29 hops.

Fig. 3(c) shows the proportion of the backup paths whose failure is caused by the path overlap. It indicates that over 50.29% of backup path failures are caused by the path overlap on average. Compared with **Fig. 3(b)**, its changing trend is consistent with the simultaneous failure probability. Although the path overlap is not only reason that leads to the simultaneous failure of the previously selected backup p the ath, it does considerably affect the effectiveness of the selected backup path. In essence, the delay-based approaches are insufficient to select a proper backup path to detour around the link failures on the default paths. The topology information is essential for identifying a more disjoint path, which motivates us to utilize the *traceroute* probing.

5. Robust Backup Path Selecting Approach

5.1. Overview of RBPS

The simultaneous failure problem motivates us to propose the RBPS approach to select a proper one-hop backup path for any node pair. In a word, it employs the *traceroute* probing in the foundation of the grid quorum system. The link-state table of each node is compressed by the Bloom filter before transmission so as to reduce its size. **Table 1** summarizes the key notations in this paper.

Table 1. Summary of the key notations

Notations	Discription
N	Network size
(i,j)	Node in the network
$Q_{i,j}$	Quorum consist of the nodes in the ith row and jth column except for node (i,j)
$\alpha_{(i,k,j)}$	Overlap ratio between the one-hop path $Path_{(i,k,j)}$ and the default path $Path_{(i,j)}$
$BF(X)$	Bloom filter encoded from path X
$Z(BF(X))$	Number of '0' bits in $BF(X)$
m	Bit length of the Bloom filter
n	Capacity of the Bloom filter
k	Number of hash functions in the Bloom filter

Suppose that there are N nodes in a full-mesh overlay network. The source and destination nodes are denoted as s and d, respectively. Node i denotes a relay node through which a one-hop candidate path forms. The default path from s to d is denoted as $Path_{(s,d)}$, while the one-hop path through the relay node i is denoted as $Path_{(s,i,d)}$.

The RBPS approach is built upon two widely used assumptions: 1) the delay of $Path_{(i,d)}$ is equal to that of $Path_{(d,i)}$; 2) the physical path between arbitrary node pairs is symmetrical, i.e., the IP sequence of $Path_{(i,d)}$ is in the reverse order of that of $Path_{(d,i)}$. Thus we have

$$Delay\left(Path_{(s,i,d)}\right) = Delay\left(Path_{(s,i)}\right) + Delay\left(Path_{(i,d)}\right). \quad (2)$$

$$Trace\left(Path_{(s,i,d)}\right) = Trace\left(Path_{(s,i)}\right) + Trace\left(Path_{(i,d)}\right). \quad (3)$$

Based on the grid quorum system and *traceroute* probing, we propose a 3-phases RBPS approach, including link-state probing, link-state table distribution and recommendation of the backup path. A $\sqrt{N} \times \sqrt{N}$ grid quorum system is achieved in advance.

Table 2. Sample of the link-state table maintained by node (1, 1)

DstID	*Delay*	*IPTrace*
(1,2)	57.39	217.149.196.50;217.149.196.51; ⋯
(1,3)	1134.32	145.99.179.145;217.149.196.50; ⋯
(1,4)	561.24	217.149.196.51;145.145.19.61; ⋯
(2,1)	342.09	145.99.19.61;145.145.80.65; ⋯
(2,2)	94.39	216.24.184.85;216.24.186.85; ⋯
…	…	…

Link-state probing. Each node actively monitors its paths to the other $N-1$ nodes in the network via *traceroute* probing periodically. When the messages return, the node maintains a link-state table locally. There are three fields in the link-state table, including *DstID*, *Delay* and *IPTrace*. The *DstID* field records the IDs of the destinations, while the *Delay* field records the delay along the default path. Besides, the *IPTrace* field records the IP address sequences on the default path. A sample of the link-state table maintained by node $(1,1)$ is shown in **Table 2**.

Link-state table distribution. Each entry in the *IPTrace* field will be encoded into a Bloom filter before the distribution (See Section 5.3). The compressed link-state table is distributed to all the nodes in the grid quorum of the current node. In that case, there are at least two rendezvous nodes that receive the link-state tables from the both sides of a node pair. The rendezvous nodes thus are able to identify a proper one-hop backup path for that node pair. For node pair consists (i,j) and (u,v), node (i,v) is selected as the preferred rendezvous node, while node (u,j) acts as a redundant.

Backup path recommendation. At the preferred rendezvous node, the one-hop candidate path through an arbitrary relay node is formed by two Bloom filters. And the overlap between each candidate path and the default path is estimated. Based on the heuristic backup path selection approach (See Section 5.2), the preferred rendezvous node (i,v) selects a proper backup path among all the candidate paths, and returns the recommendation message to the nodes (i,j) and (u,v).

The grid quorum in **Fig. 2** is taken to illustrate the RBPS approach. For node pair consists of $(1,4)$ and $(4,1)$, node $(1,4)$ sends a *traceroute* probing message to all the other 15 nodes in the network every 15 minutes. When node $(1,4)$ receives the returned messages, it will immediately add a new entry to its link-state table. If the entry already exists, it will update the current entry. Node $(1,4)$ then sends its link-state table, compressed by the Bloom filter, to all of nodes in $Q_{1,4} = \{(1,1),(2,1),(2,3),(4,2),(4,3),(4,4)\}$. Similarly, node $(4,1)$ sends its compressed link-state table to all of nodes in $Q_{4,1}$. Finally, node $(1,1)$ receives the link-state tables from both nodes $(4,1)$ and $(1,4)$, and hence can identify a proper one-hop backup path for the node pair. Node $(1,1)$ is preferred as a rendezvous node, and sends a recommendation message back to the node pair. In a global scope, node $(1,1)$ sends one-hop backup path recommendation messages to all nodes in $Q_{1,1}$. In this way, all of the node pairs in this network receive the recommendation messages. When the default path fails, the node pair is able to communicate along such a one-hop backup path to detour around the failed links.

5.2. Heuristic Backup Path Selection

As we have mentioned above, selecting the backup path merely considering of the delay is not feasible. If there is a one-hop path, whose delay and the overlap with the default path are the

least at the same time, is definitely a desired backup path. The two criteria, however, are rarely satisfied in pair. The path selected based on one criterion may exhibit poor performance in terms of the other criterion. Due to the simultaneous failure problem, reducing the overlap between the backup and default path is taken as the primary objective to improve the end-to-end communication performance. If the backup path is unavailable, it is useless no matter how low the delay is. On this basis, we propose a heuristic approach to leverage the two criteria, i.e., selecting the one-hop backup path with the minimum delay among the candidate paths whose overlap ratio is under a certain threshold.

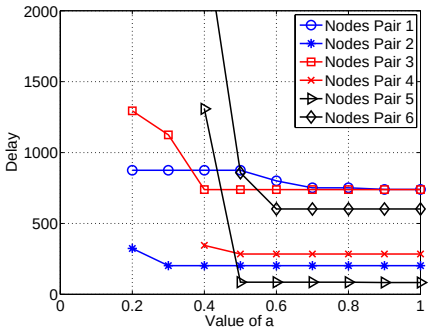
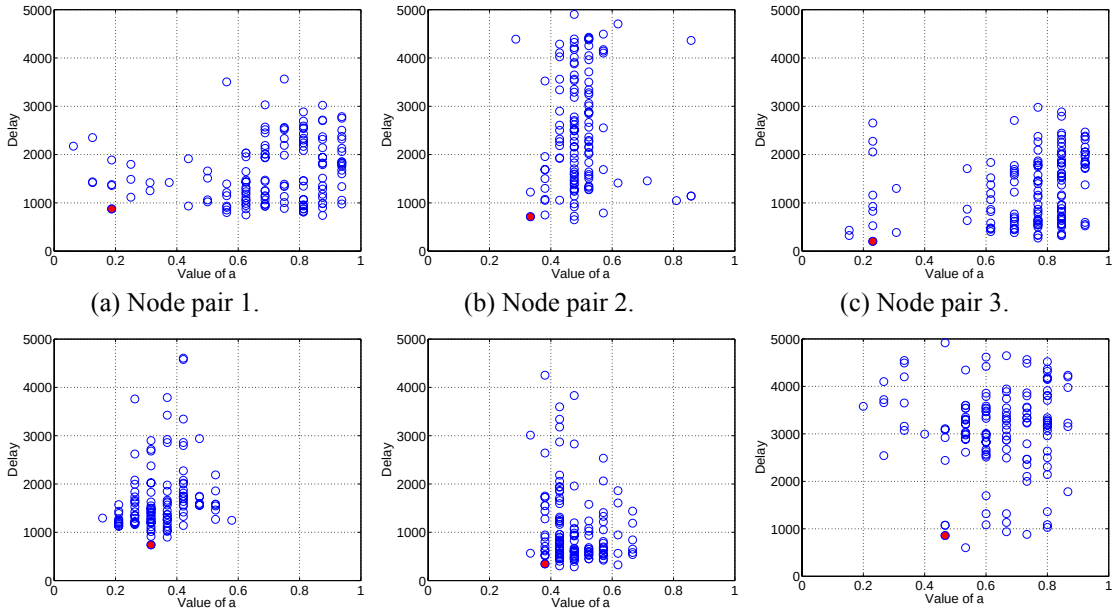


Fig. 5. The changing trend of delay as the increase of the overlap ratio for six randomly selected node pairs.

To eliminate the influence of path length, the overlap ratio between the candidate path $Path_{(i,k,j)}$ and the default path $Path_{(i,j)}$, denoted as $\alpha_{(i,k,j)}$, is defined as a criterion to represent the level of path overlap. It can be computed by Equation (4), where $Length\ (Pat\ h_{(i,j)})$ is the length of $Pat\ h_{(i,j)}$, while $Overlap(Pat\ h_{(i,k,j)}, Pat\ h_{(i,j)})$ is the length of the overlaps between $Pat\ h_{(i,k,j)}$ and $Pat\ h_{(i,j)}$.

$$\alpha_{(i,k,j)} = \frac{Overlap(Path_{(i,k,j)}, Path_{(i,j)})}{Length\ (Path_{(i,j)})} \tag{4}$$



(a) Node pair 1. (b) Node pair 2. (c) Node pair 3.

(d) Node pair 4. (e) Node pair 5. (f) Node pair 6.

Fig. 6. Delay and overlap ratio distribution of the one-hop paths for six randomly selected node pairs.

Although a more similar one-hop path may show more similar performance, there are still some paths that are more disjoint with the default one, while its delay is relatively less. To support such a hypothesis, the actual link state between each node pair is investigated via extensive observations. We derive a 10-days dataset, including 166,530 node pairs, from iPlane dataset and analyze the changing of delay along with the increase of the overlap ratio.

In a global view, we compare the delay gain under different overlap ratio upper bound α, i.e., let $\alpha = 0.1, 0.2, \ldots 0.9$. The delay gain is a measurement of the delay difference between the backup path selected by our approach and the SAP approach [5]. Actually, when α is set to be 1.0, the selected one-hop backup path is equivalent to the path selected by the SAP approach.

For the ease of illustration, we randomly choose six node pairs to show the relationship between the delay and overlap of all the one-hop paths. The results under more node pairs are quite similar. For each node pair, the minimum delays under different α are plotted in **Fig. 5**. When α achieves a certain value, the minimum delay of the selected path will not reduce any more. For these node pairs, the distribution of both the delay and overlaps between these candidate paths and the default path are plotted in **Fig. 6**, where the red points represent the proper backup paths that we desire to select for these node pairs. Obviously, the delay of these backup paths is near to the minimum delay, while their overlap ratio is quite smaller than the minimum overlap ratio of all the paths. In summary, most of the node pairs in our observations satisfy the following three rules.

1. For any node pair, the one-hop path with the maximum overlap ratio does not exhibit the minimum delay. When the overlap ratio upper bound α grows, the delay of the selected backup path increases for 70.26% of the node pairs.
2. Compared with the one-hop path with the minimal delay, there are 92.73% of all node pairs that has a path that overlaps with the default path less, while its delay is near to the minimum value among all the candidates (1.2 times of the minimum value). Therefore, it is not necessary to enlarge the value of α to derive the proper backup path.
3. When the upper bound of the overlap ratio α is set to be in the interval [0.4,0.5], there exists a more proper candidate path for 79.83% of the node pairs.

The above laws provide direct evidences to support our backup path selection scheme, as well as the parameter selection of α.

5.3. Compressing Link-state Table with Bloom Filters

The link-state table with raw IP level path information is $160\sqrt{(N-1)^3}$ bytes in average (16 hops of each entry to $2\sqrt{N-1}$ nodes), which is too large to be distributed directly. This paper further reduces the size of the link-state table with the Bloom filter. The Bloom filter we utilized is a m bits vector with k independent hash functions. A path X can be presented as a Bloom filter through a mapping relationship: $X \rightarrow BF(X)$. Note that, the hash functions employed by the Bloom filters are assumed to be all independent, as demonstrated in [14]. The process of encoding an IP sequence to a Bloom filter is quite straightforward. This paper focuses on two fundamental operations for identifying a proper backup path, i.e., one-hop candidate path formation and overlap estimation between the default and backup paths. Both of them are operated at the rendezvous node, involving two independent Bloom filters.

5.3.1. One-hop Path Formation

Before we further propose the formation of the one-hop path, it is necessary to clarify the algebraic operation of Bloom filters. In our previous work [15], we have defined both the union and intersection operations of Bloom filters, and quantified the uncertainty in

approximating the set union/intersection with the Bloom filter union/intersection. The Bloom filters mentioned in the remainder of this paper use the same m and hash functions, i.e., each node assign a same group of hash functions and the m bit length for each Bloom filter. The demonstrations of Theorem 1 and Theorem 2 are omitted due to page limitation. More details can be found in [15].

Definition 2. (Union of Bloom filters) Assume that $BF(A)$ and $BF(B)$ use the same m and hash functions. Then, the union of $BF(A)$ and $BF(B)$, denoted as $BF(B))$, can be represented by a logical *OR* operation between their bit vectors.

Theorem 1. If $BF(A \cup B)$, $BF(A)$ and $BF(B)$ use the same m and hash functions, then

$$BF(A \cup B) = BF(A) \cup BF(B). \tag{5}$$

Definition 3. (Intersection of Bloom filters) Assume that $BF(A)$ and $BF(B)$ use the same m and hash functions. Then, the intersection $BF(A)$ and $BF(B)$, denoted as $BF(C)$, can be represented by a logical *AND* operation between their bit vectors.

Theorem 2. If $BF(A \cap B)$, $BF(A)$, and $BF(B)$ use the same m and hash functions, then

$$BF(A \cap B) = BF(A) \cap BF(B). \tag{6}$$

with probability $(1 - 1/m)^{k^2|A-A\cap B||B-A\cap B|}$.

A path can be viewed as a set of IP address. The one-hop candidate path consists of the IP address on the default path from the source node s to the relay node i, and the default path from the relay node i to the destination node d. It can be represented by the union of two sets. Let $BF(Path)$ denote the path encoded by a Bloom filter, according to Equation (5), the candidate path can be represented by Equation (7).

$$BF(Path_{(s,i,d)}) = BF(Path_{(s,i)}) \cup BF(Path_{(i,d)}). \tag{7}$$

Actually, the one-hop candidate path formation is a union operation of two Bloom filters that use the same m and hash functions. At the rendezvous node of a node pair, $N - 2$ one-hop candidate paths are formed through $N - 2$ relay nodes.

5.3.2. Path Overlap Estimation

The estimation of the path overlap is challenging, when the default and one-hop paths are encoded into two Bloom filters. Due to the uncertainty in the intersection operation of Bloom filters, as illustrated in Theorem 2, the path overlap cannot be estimated by the intersection of two Bloom filters directly. Besides, the Bloom distance in [16] is a criterion of path similarity, rather than the accurate estimation of path overlap length. On this basis, an accurate estimation of the path overlap is further proposed.

For any Bloom filter $BF(X)$, the size of set X can be estimated by Equation (8) [14]. After all of the n elements of X have been hashed into a Bloom filter, the probability of that the jth bit is $'0'$ is $(1 - 1/m)^{k|X|}$. Let $Z(BF(X))$ denote the number of $'0'$ bits in $BF(X)$. The mathematical expectation of $Z(BF(X))$ is concentrated around $m(1 - 1/m)^{k|X|}$. Therefore, we can estimate the size of X by Equation (8). Note that, the value of Let $Z(BF(X))$ is obtained by counting the number of "0" bits in the Bloom filter.

$$|X| = \frac{\ln(Z(BF(X))) - \ln m}{k \ln(1 - 1/m)}. \tag{8}$$

The path overlap can be represented by the intersection of two sets that represent the default and backup paths, respectively. Equations (8) indicates that $|A \cap B|$ can be estimated by $BF(A \cap B)$. The uncertainty of the intersection operation of Bloom filters, however, indicates that $BF(A \cap B)$ cannot be obtained by the intersection of $BF(A)$ and $BF(B)$ directly. Note that, the size of A, B, $A \cap B$ and $A \cup B$ satisfy that

$$|A \cap B| = |A| + |B| - |A \cup B|. \tag{9}$$

According to Equation (5), it is more accurate to estimate the size of the union of two sets. Therefore, the question can be transformed to estimate $|A \cup B|$, rather than $|A \cap B|$. We therefore propose the path overlap estimation approach with Bloom filters in Theorem 3.

Theorem 3. Given Bloom filters $BF(A)$, $BF(B)$ and $BF(A \cup B)$, which use the same m and hash functions, the size of $|A \cap B|$ can be estimated by Equation (10).

$$|A \cap B| = \frac{\ln\big(Z(BF(A))\big) + \ln\big(Z(BF(B))\big) - \ln\big(Z(BF(A) \cup BF(B))\big) - \ln m}{k \ln(1 - 1/m)}. \tag{10}$$

Proof. For these Bloom filters, the number of $'0'$ bits can be obtained. Therefore, the size of A, B and $A \cup B$ can be estimated by Equation (8). We can derive that

$$|A| = \frac{\ln\big(Z(BF(A))\big) - \ln m}{k \ln(1 - 1/m)}. \tag{11}$$

$$|B| = \frac{\ln\big(Z(BF(B))\big) - \ln m}{k \ln(1 - 1/m)}. \tag{12}$$

$$|A \cup B| = \frac{\ln\Big(Z\big(BF(A \cup B)\big)\Big) - \ln m}{k \ln\left(1 - \frac{1}{m}\right)} = \frac{\ln\big(Z(BF(A) \cup BF(B))\big) - \ln m}{k \ln\left(1 - \frac{1}{m}\right)} \tag{13}$$

According to the relationship among the size of A, B, $A \cap B$ and $A \cup B$ in Equation (9), we can obtain the estimation of $|A \cap B|$ as shown in Equation (10). ■

Actually, if $BF(A \cap B)$ is given, the size of $A \cap B$ can be estimated by

$$|A \cap B| = \frac{\ln\Big(Z\big(BF(A \cap B)\big)\Big) - ln\, m}{k\, ln(1 - 1/m)}. \tag{14}$$

Again, we substitute the above Equation to Equation (10), and thus obtain Corollary 1, which connects the proportions of the $'0'$ bits of two Bloom filters and their union and intersection.

Corollary 1. Given Bloom filters $BF(A)$, $BF(B)$, $BF(A \cap B)$ and $BF(A \cup B)$ with the same number of bits and using the same hash functions. Then the number of the "0" bits in these four Bloom filters satisfy Equation (15).

$$\ln Z\big(BF(A \cap B)\big) = \ln Z(BF(A)) + \ln Z(BF(B)) - \ln Z(BF(A) \cup BF(B)). \tag{15}$$

For any node pair, let A and B denote the two parts of the one-hop candidate path, and let C denote the default path. The overlap between the default and candidate paths can be estimated as follows.

$$|A \cup B \cap C| = \frac{\ln\big(Z(BF(A) \cup BF(B))\big) + \ln(Z(BF(C))) - \ln\big(Z(BF(A) \cup BF(B) \cup BF(C))\big) - \ln m}{k \ln(1 - 1/m)}. \tag{16}$$

Note that, the path overlap estimation can be done just by executing the union operation twice. One is to obtain the union of $BF(A)$ and $BF(B)$, while the other one is to obtain the

union of $BF(A \cup B)$ and $BF(C)$. As we have mentioned in Equation (5), the union operation can be done with no error. The accuracy of the path overlap estimation is greatly improved.

The communication overhead of the RBPS approach is bound by $O(n^{1.5})$, as indicated in Theorem 4. Each *traceroute* probing message requires 8 bytes. For each entry in the link-state table, 4 bytes are assigned to store each IP address, while 1 byte is assigned to store the delay. The average length of end-to-end paths in the collected iPlane dataset is 15.76. We thus assume that the path length is 16 to estimate the communicating overhead in RBPS. To avoid confusion, we have to emphasize that the notation N denotes the number of nodes in the overlay network, rather than n that denotes the capacity of the Bloom filter. Actually, N and n are two independent parameters; the value of n depends on the maximum of hops along a one-hop path.

Theorem 4. Identifying the best one-hop backup path in RBPS makes every node incur $N + 4\sqrt{N} - 3$ messages sizing of $\frac{m}{4}(N-1)(\sqrt{N}-1) + 8(N-1) + 16(\sqrt{N}-1)$ bytes in total, where m denotes the bit length of the Bloom filter and N denotes the number of nodes in the overlay network.

Proof. First of all, any node (i, j) probes its link-state to all the other $N-1$ nodes in the network. Such a process generates $N-1$ messages sizing of $8(N-1)$ bytes in total. Each entry in the link-state table requires m bytes in average. As a result, a link-state table sizing of $\frac{m}{8}(N-1)$ bytes for each node is formed to distribute to the other nodes. Furthermore, node at (i, j) sends its link-state table to all nodes in $Q_{i,j}$,which results in $(\sqrt{N}-1)$ messages sizing of $\frac{m}{4}(N-1)(\sqrt{N}-1)$ bytes in total. Finally, node (i, v) sends the routing recommendation messages back to all the nodes in $Q_{i,v}$, hence causing $2(\sqrt{N}-1)$ messages sizing of $16(\sqrt{N}-1)$bytes in total. Therefore, the total communication overhead is $N + 4\sqrt{N} - 3$ messages sizing of $\frac{m}{4}(N-1)(\sqrt{N}-1) + 8(N-1) + 16(\sqrt{N}-1)$ bytes. ■

To the best of our knowledge, we are the first to propose the path overlap estimation with Bloom filters to measure the actual overlap length between two paths. This approach can be easily extended to any other similar context. Besides, the effectiveness of our RBPS approach relies on the design of the Bloom filter that encodes the IP address sequence on the default path. The parameters m, n and k are tuned to minimize the false positive rate. This paper discusses the design of Bloom filter via trace-driven experiments in Section 5.2.

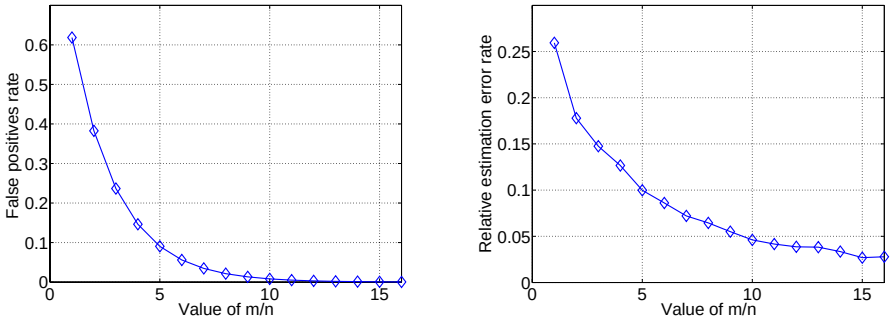
5. Experimental Study

This paper proposes the RBPS approach to improve the robustness of the selected one-hop path, and further employs the Bloom filter to considerably reduce the communication overhead. To evaluate the feasibility and performance of our approaches, we conduct extensive trace-driven emulations based on the iPlane dataset, in respects of the accuracy of path overlap estimation, simultaneous failure probability, delay gain, and communication overhead. The current best delay-based approach SAP [5] is taken as a reference.

5.1. Accuracy of Path Overlap Estimation

False positive in the Bloom filter suggests that an element x is in X even though it is not. It may induce errors in the path overlap estimation. In our experimental settings, we fix the capacity n of the Bloom filter as the maximum length of the one-hop paths, i.e., 24, and set

$m/n = 1, 2, \ldots, 16$. According to Equation (1), k is set to be $(m/n)\ln 2$ for minimizing the false positive rate. **Fig. 7(a)** plots the changing trend of the false positive rate along with the value of m/n. When the value of m/n grows, the false positive rate decreases. When the value of m/n is beyond 10, the false positive rate is below 0.01 and decreases smoothly.



(a) False positive rate. (b) Relative estimation error rate.

Fig. 7. False positive rate and relative estimation error rate of path overlap estimation.

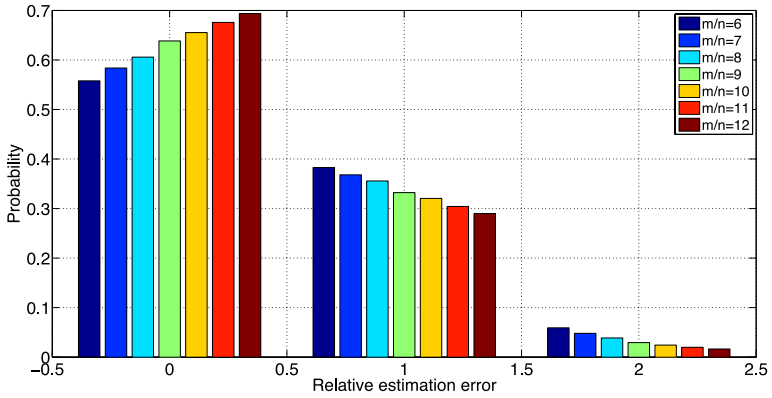


Fig. 8. Probability distribution of relative estimation error under different m/n.

This paper further defines the relative estimation error as a criterion of path overlap estimation accuracy. It is the difference between the estimated path overlap length and the real length. As the path overlap is an integer, the estimated value should be rounded to the nearest integer. Therefore, the estimation whose relative estimation error is less than 0.5, will be viewed as the same with the actual path overlap value. In the experiment, the overlap between all of the 181 candidate paths and the default path for each of the 183×182 node pairs are measured. We then compare the estimated value and the real value to obtain the relative estimation error. **Fig. 7(b)** displays the changing trend of the relative estimation error rate along with the value of m/n. When the value of m/n grows, the relative estimation error rate decreases. When the value of m/n is beyond 10, the false positive rate is below 0.05 and gets smoother. When $m/n = 15$, the relative estimation error rate achieves its minimum value 0.0268.

Fig. 8 plots the probability distribution of the relative estimation error under different m/n. After rounding the value of the relative estimation error, its distribution is concentrated at 0, 1 and 2, respectively. According to the definition of relative estimation error, the probability density of the relative estimation error concentrated around 0 directly illustrates the accuracy of path overlap estimation based on Bloom filters. The probability density in such an interval is significantly larger than that in the other intervals. Although the probability density grows

along with the increase of m/n, it becomes steady after the value of m/n reaches 10, which is in agreement with the conclusion in **Fig. 7**.

5.2. Parameter Tuning of Bloom filter

Optimized parameters of the Bloom filter can improve the performance of our RBPS approach. Based on the requirement of path overlap estimation, we derive three constraints on the parameters:

1) The false positive rate should be below 0.01 for any single path overlap estimation.
2) The relative estimation error rate should be no more than 0.1.
3) The probability that the relative estimation error ranges in $[-0.5, 0.5]$ should be as large as possible.

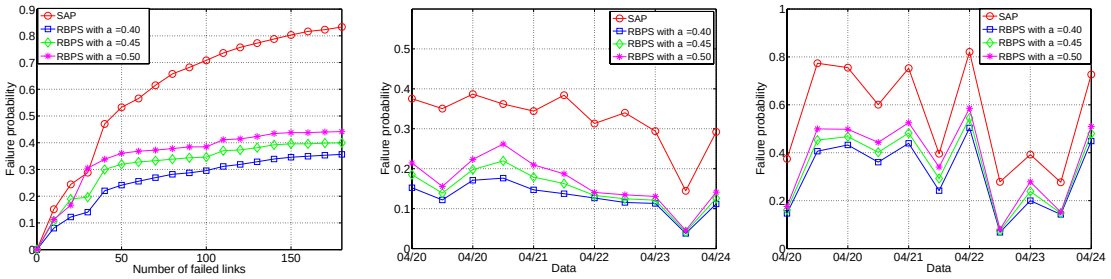
Fig. 7 and **Fig. 8** provide evidences for the determination of such parameter settings. The capacity of the Bloom filter, i.e., n, should be an upper bound of the maximum hops of the one-hop candidate path. In our trace-driven experiments, all of the lengths of the candidate paths are below 24. To minimize the false positive rate, the number of hash functions k is set to be $(m/n)\ln 2$. The remaining parameters can be determined according to the above constraints. When $m/n = 10$, the false positive is 0.008192, the average relative estimation error is 0.04607, and the probability density in interval [−0.5, 0.5] is large enough and progressively steady. Therefore, we have $m = 240$, $k = 6$, and $n = 24$.

5.3. Simultaneous Failure Probability

It is the simultaneous failure problem in our observation that motivates us to select a more disjoint backup path to improve the fault-tolerant ability. Extensive simulations are conducted to evaluate the simultaneous failure probability in the RBPS approach. The links on the default paths between C_n^2 node pairs are selected randomly to be failed to simulate the link failures on the Internet. Besides, along with the changing of the overlap ratio upper bound α, we can obtain different one-hop paths to take over the failed default path. In Section 5.2, the range of α is set in $[0.40, 0.50]$. Here we suppose that the value of α is 0.40, 0.45 and 0.50 in the experimental setting, respectively.

We first study the effect of the network state on the simultaneous failure problem. **Fig. 9(a)** shows the simultaneous failure probability with different link failure number, ranging from 10 to 180 (form 0.06% to 1.08% of all the default paths), on the same day. Compared with the SAP approach, the fault-tolerant ability is significantly improved. When the number of failed links is small, the difference between the different approaches is not significant. When such a number is beyond 40, the simultaneous failure probability in our RBPS approach is much smaller than that in the SAP approach. When the failure probability of the selected backup in the delay-based approach is over 80%, such a probability in our RBPS approach are below 40%, irrespective of the value of α. A smaller α may derive a more robust backup path.

To validate the flexibility of our approaches, we compare the difference between the simultaneous failures on different days, as plotted in **Fig. 9 (b)**. A 11-days (from April, 20th, 2011 to April, 30th 2011) iPlane dataset is chosen, and the number of the failed links is set to be 40 constantly, which is the inflection point in **Fig. 9(a)**. There are 0.24% of all the default paths fails. The backup paths selected by the RBPS approach exhibit lower simultaneous failure probability. When $\alpha = 0.45$, the simultaneous failure probability for the RBPS is 60% smaller than that for the SAP approach, on average.



(a) Simultaneous failure probability with different number of failed links on the same day.

(b) Simultaneous failure probability on different day with 40 failed links.

(c) Simultaneous failure probability on different day with different number of failed links.

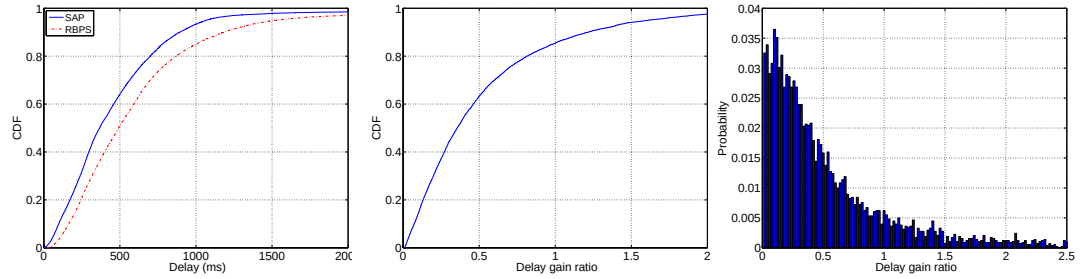
Fig. 9. Comparison of the simultaneous failure probability in different approach and different m/n.

Furthermore, we conduct a more practical experiment based on the 11-days dataset. The number of failed links is a random variable ranging from 0 to 180 to simulate the state of the network in real large-scale network applications. The one-hop backup path between each node pair is selected based on the link state on different days. As shown in **Fig. 9(c)**, our RBPS approach identifies a more robust backup path that can detour around the failed links on the default path. When the state of the network gets worse, the advantage of the RBPS approach is more significant.

In summary, the experimental results indicate that our RBPS approach overweighs the delay-based approach in respect of fault-tolerant ability, regardless of the state of the network and the selection of the overlap ratio upper bound. A smaller α can identify a more robust backup path.

5.4. Delay Gain of the Selected One-hop Backup Path in Our Approaches

The end-to-end communication performance is finally reflected in one thing: the delay. We desire to identify a more disjoint path without sacrificing the delay. The delays under the SAP and RBPS are compared based on the iPlane Dataset, after removing the failed path. To clarify, we observe that the value of α has slight influence on the delay of the selected backup path. We plot the CDFs under different values of α at the first time. Those curves, however, almost coincide with each other. For convenience, we fix α to be 0.40 to compare the delay performance of different approaches.



(a) CDFs of delay.

(b) CDF of delay gain ratio.

(c) Frequency histogram of delay gain ratio.

Fig. 10. Delay Comparison between the RBPS approach and SAP approach.

Fig. 10(a) shows the CDFs of the delays for the SAP and RBPS approaches. Overall, the delay of SAP is smaller than that of RBPS, but not much. We measure the delay gain ratio, the ratio of the delay difference to the delay in SAP, to compare the delay difference for each node

pair. **Fig. 10(b)** shows the CDFs of the delay gain ratio for the RBPS. The delay gain ratio of over 62 percent of the alternative paths remains less than 0.5, and over 85 percent remains less than 1. **Fig. 10(c)** plots the probability distribution histogram in the RBPS. Most of the delay gain ratios are concentrated near 0.14.

The delays of the one-hop paths selected by our approach, are not much larger than that of the delay-based selection approach. The failed paths in both the RBPS and the SAP approach are removed. Furthermore, the improvement of fault-tolerant ability overweighs the tradeoff of the slight increment of delay.

5.5. Communication Overhead

The communication overhead directly affects the scalability of our approaches. In this section, the per-node communication overhead of our approaches is evaluated, compared with the RON [3] and the SAP [5] approach. The size of the link-state table that contains the IP path is much larger than that contains delay information only.

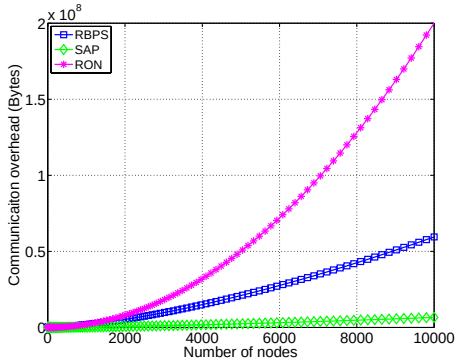


Fig. 11. Overhead Comparison among RBPS (m=240), SAP and the RON approaches.

The bit length of the Bloom filters in our RBPS approach is set to be 240, according to the parameter discussion in Section 5.2. The experimental results are plotted in **Fig. 11**. Our RBPS approach indeed dramatically reduces the per-node communication overhead by about 60%, compared with the RON, which employs all-pairs ping probing, distributes the link-state table to all the other nodes and selects the best one-hop backup path merely based on the delay criterion, irrespective the size of the network. Additionally, the RBPS shows similar scalability performance, compared with the currently least overhead SAP approach. When the network size is 1000, the RBPS incurs about 1MB extra per-node communication overhead.

The communication overhead is a necessary trade-off of the improvement failure-tolerant ability. Actually, the per-node overhead produced by the RBPS is bounded by $O(N^{1.5})$, as we have proved in Theorem 4. Nevertheless, its fault-tolerant ability is significantly better than that of the SAP approach, which overweighs the drawback in communication overhead. Besides, such a communication overhead can be afforded in the current large-scale systems in practice.

6. Conclusion

Path diversity in overlay networks is important for improve the end-end communication performance for delay-sensitive network applications. The delay-based approaches, however, suffer from the simultaneous failure problem caused by the correlation between the default and backup paths, while the topology-based approaches are either lack of certainty or incurring

scalability limitation. This paper focuses on identifying a proper backup path to bypass the failed links on the default path for each node pair in an overlay network. Our long-term observations show that the simultaneous failure problem occurs too frequently. The root cause of such a problem is the overlap between the selected backup path and the default path. To address such a challenging issue, we propose RBPS approach that can derive a more disjoint one-hop backup path while its delay is near to the lowest. Our RBPS approach employs traceroute probing to obtain fine-grained topology information, and utilizes the grid quorum system and Bloom filters to reduce the communication overhead. We propose a heuristic backup selection to leverage the delay and overlap to identify a more proper backup path.

Extensive trace-driven evaluations show that the RBPS approach is able to select a more robust backup path to bypass the link failures. The simultaneous failure probability is reduced by about 60%, while the delay gain ratio of the selected backup paths is concentrated on 14%. Additionally, the per-node communication overhead of the two methods is bounded by $O(N^{1.5})$. Comparing with the approaches based on the ping probing, our RBPS approach shows a similar scalability performance with the current best SAP approach. In summary, our RBPS approach significantly improves the performance on each criterion, and therefore better supports the large-scale distribution of the delay-sensitive network applications.

Following this paper, we intend to extend our current work to further address the issue of path overlap in the future. An IP-level path is assumed to be symmetrical in order to simplify the problem, which is not always true in practice. Therefore, we will further propose another backup path selecting approach with the reverse *traceroute* [20] in the future works. Furthermore, the grid Quorum System will be extended to a higher dimension to acquire more desirable properties to improve the scalability in large-scale distributed systems.

References

[1] J. Liao, J. Wang, T. Li and X. Zhu, "Introducing multipath selection for concurrent multipath transfer in the future Internet," *Computer Networks*, vol. 55, no.4, pp. 1024–1035, March, 2011. Article (CrossRef Link).

[2] S. Tao, K. Xu, A. Estepa, T. Fei, L. Gao, R. Gurin, J. F. Kurose, D. F. Towsley and Z. L. Zhang, "Improving voip quality through path switching," in *Proc. of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, pp. 2268–2278 vol. 4, March 13-17, 2005. Article (CrossRef Link).

[3] D. Andersen, H. Balakrishnan, F. Kaashoek and R. Morris, "Resilient Overlay Networks," in *Proc. of 18th ACM Symposium on Operating Systems Principles (SOSP)*, pp. 131–145, October 21-24, 2001. Article (CrossRef Link).

[4] K. P. Gummadi, H. V. Madhyastha, "Improving the Reliability of Internet Paths with One-hop Source Routing," in *Proc. of 6th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, pp. 183-198, December 5, 2004. Article (CrossRef Link).

[5] D. Sontag, Y. Zhang, A. Phanishayee, D. G. Andersen, D. Karger, "Scaling All-Pairs Overlay Routing," in *Proc. of the 5th international conference on Emerging networking experiments and technologies (CoNext)*, pp. 145-156, December 1-4, 2009. Article (CrossRef Link).

[6] T. Fei, S. Tao, L. Gao and R. Guerin, "How to Select a Good Alternate Path in Large Peer-to-Peer Systems?," in *Proc. of the 25th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, pp. 1106-1118, April 23-29, 2006. Article (CrossRef Link).

[7] M. Luckie, Y. Hyun and B. Huffaker, "Traceroute probe method and forward IP path inference," in *Proc. of the 8th ACM SIGCOMM conference on Internet measurement*, pp. 311-324, 2008. Article (CrossRef Link).

[8] B. Y. Zhao, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph and J. D. Kubiatowicz, "Tapestry: A

Resilient Global-scale Overlay for Service Deployment," *IEEE Journal on Selected Areas In Communications*, vol. 22, no. 1, pp. 41-53, Jan. 2004. Article (CrossRef Link).

[9] M. Zhang, J. Lai, A. Krishnamurthy, R. Wang and L. Peterson, "A Transport Layer Approach for Improving End-to-End Performance and Robustness Using Redundant Paths," in *Proc. of USENIX the annual conference on USENIX Annual Technical Conference,* pp. 99-112, Jun 27- July 2, 2004. Article (CrossRef Link).

[10] A. Nakao, L. Peterson and A. Bavier, "A Routing Underlay for Overlay Networks," in *Proc. of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM)*, pp. 11-18, August 25-29, 2003. Article (CrossRef Link).

[11] M. Vukolić, "Quorum systems: With applications to storage and consensus," *Synthesis Lectures on Distributed Computing Theory*, vol. 3, no. 1, pp. 1-146, Feb. 2012. Article (CrossRef Link).

[12] M. Naor and U. Wieder, "Scalable and dynamic quorum systems", *Distributed Computing*, vol. 17, no. 4, pp. 311-322, May, 2005. Article (CrossRef Link).

[13] B. H. Bloom, "Space/Time Trade-offs in Hash Coding with Allowable Errors," *Communications of the ACM*, vol. 13 no. 7, pp. 422-426, July, 1970. Article (CrossRef Link).

[14] A. Broder and M. Mitzenmacher, "Network Applications of Bloom Filters: A Survey," *Internet Mathematics*, vol. 1, no. 4, pp. 485-509, January, 2004. Article (CrossRef Link).

[15] D. Guo, J. Wu, H. Chen and X. Luo, "Theory and Network Applications of Dynamic Bloom Filters," in *Proc. of the 25th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, pp. 1-12, April 23-29, 2006. Article (CrossRef Link).

[16] B. Donnet, B. Gueye and M. A. Kaafar, "Path similarity evaluation using Bloom filters," *Computer Networks*, vol. 56, no. 2 pp. 858-869, Feb. 2012. Article (CrossRef Link).

[17] All-pairs-ping dataset, [Online]. Available: http://pdos.csail.mit.edu/~strib/_pl_app/.

[18] iplane dataset, [Online]. Available: http://iplane.cs.washington.edu/.

[19] H. V. Madhyastha, T. Isdal, M. Piatek, C. Dixon, T. Anderson, A. Krishnamurthy and A. Venkataramani, "iPlane: An Information Plane for Distributed Services," in *Proc. of USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, pp.367-380, November 6-8, 2006. Article (CrossRef Link).

[20] E. Katz-Bassett, H. V. Madhyastha, V. K. Adhikari, C. Scott, J. Sherry, P. van Wesep, T. Anderson, A. Krishnamurthy, "Reverse traceroute," in *Proc. of the 7th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, pp. 15–31, April 28-30, 2010. Article (CrossRef Link).

[21] D. Guo, H. Jin, T. Chen, J. Wu, L. Lu, D. Li, and X. Zhou, "Partial Probing for Scaling Overlay Routing," *Online published at IEEE Transactions on Parallel and Distributed Systems*, November 29, 2012. Article (CrossRef Link).

[22] 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 and Distributed Systems*, vol. 22, no. 6, pp. 1042-1055, June, 2011. Article (CrossRef Link).

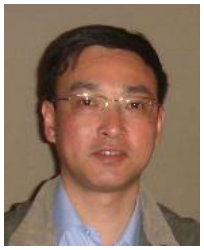
[23] D. Guo, Y. Liu, H. Jin, Z. Liu, and W. Zhang, "Theory and Network Applications of Balanced Kautz Tree Structures," *ACM Transactions on Internet Technology*, vol. 13 no. 1, Article No. 3, June, 2012. Article (CrossRef Link).

[24] D. Guo, Y. Liu and X. Li, "BAKE: A Balanced Kautz Tree Structure for Peer-to-Peer Networks," in *Proc. of the 27th IEEE International Conference on Computer Communications*, April 13-18, 2008. Article (CrossRef Link).

Xiaolei Zhou received the BA degree from the Information Management Department, Nanjing University, P.R. China, in 2009, and the MS degree in military science from the National University of Defense Technology, P.R. China, in 2011. He is currently working toward the PhD degree in the School of Information System and Management, National University of Defense Technology, P.R. China. His current research interests include wireless sensor networks, Internet of things, and data center networking.

Deke Guo received the B.S. degree in industry engineering from Beijing University of Aeronautic and Astronautic, Beijing, China, in 2001, and the Ph.D. degree in management science and engineering from National University of Defense Technology, Changsha, China, in 2008. 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.

Tao Chen received the B.S. degree in military science, the MS and Ph.D. 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.



Xueshan Luo received the B.E. degree in information engineering from Huazhong Institute of Technology, Wuhan, China, in 1985, the M.S. and Ph.D. degrees in system engineering from National University of Defense Technology, Changsha, China, in 1988 and 1992, respectively. He was a faculty member and associate professor at National University of Defense Technology from 1992 to 1994 and from 1995 to 1998, respectively. Currently, he is a professor of Information System and Management, National University of Defense Technology. His research interests are in the general areas of information system and operation research. His current research focuses on architecture of information system.