Path Probing Relay Routing for Achieving High End-to-End Performance

Chen-Mou Cheng[1], Yu-Sheng Huang[2], H.T. Kung[1], and Chun-Hsin Wu[2,3]

[1] Division of Engineering and Applied Sciences
Harvard University
Cambridge, MA, USA
{doug, htk}@eecs.harvard.edu

[2] Institute of Information Science
Academia Sinica
Taipei, Taiwan
{hus,wuch}@iis.sinica.edu.tw

[3]Department of Computer Science and Information Engineering
National University of Kaohsiung
Kaohsiung, Taiwan
wuch@nuk.edu.tw

***Abstract*—We present an overlay network routing scheme, called Path Probing Relay Routing (PPRR), which is capable of promptly switching to alternative paths when the direct paths provided by the underlying IP networks suffer from serious performance degradation or outage. PPRR uses a randomized search algorithm to discover available alternative paths and employs an end-to-end, on-demand probing technique to determine their quality. To assess the effectiveness of PPRR, we conduct performance simulations using four sets of real-world traces, collected by various research groups at different times and places. Our simulation results show that the performance of PPRR is comparable to that of a typical link state relay routing algorithm. Compared with the latter, PPRR has lower probing overhead in the sense that the overhead remains constant as network size grows. In particular, PPRR avoids the need to flood the overlay network with link state updates.**

Keywords-overlay networks; relay routing; path probing; end-to-end performance

I. INTRODUCTION

Scalability is a critical issue in large-scale networks such as the Internet today. Currently, the Internet utilizes BGP [5] to exchange reachability information of coagulated groups of network nodes, based on which IP packets are routed to respective destination nodes. However, insufficient or untimely exchange of routing information resulting from a large number of network nodes may reduce reliability and efficiency of routing as perceived by applications at end nodes. The situation becomes even more complicated when the self-interests of the numerous self-administrated autonomous systems (AS) are taken into this picture: when serving as a transit AS, it may have little incentive to improve such applications' perception if there are no service agreements between the AS and the applications. As a result, the current Internet often experiences a low routing efficiency in the sense that many sub-optimal paths are used instead of optimal paths. For example, as reported in [6], for about 50% of the paths measured, there exists an alternative route with lower latency, and for almost 80% of the paths, there is an alternative path with a lower packet loss rate.

This work was supported in part by Academia Sinica and Industrial Technology Research Institute in Taiwan, which hosted the visits of Chen-Mou Cheng and H. T. Kung in 2003.

Overlay networks are an approach for relieving the aforementioned routing inefficiency [1]. Overlay networks are networks constructed over another set of networks. One overlay hop may consist of many hops in the underlying network. From such a viewpoint, the Internet provides generic connectivity to hosts in different AS's, which, under the restriction of the underlying IP networks, may form an overlay network to provide specific services, such as improving reliability or efficiency of routing by relaying packets for each other. An overlay network can provide end hosts with a means to have some control of packet routing over the Internet. The overlay routes can be specific to an application, or even to a TCP/UDP flow. This implies that a node participating in an overlay network can deal with congestion more promptly. In addition, one can use alternative or redundant paths to improve reliability or efficiency with a much finer granularity of control than what can be provided by the underlying IP networks.

Most existing overlay networks use link state routing algorithms, which in turn use link probing and network flooding to obtain and distribute the link performance information of the overlay network itself [1][2][6]. These algorithms work roughly as follows. Each participating node of the overlay network constantly monitors the quality of the overlay links to other nodes. Subsequently, the nodes flood the network to disseminate this information to every node, based on which routing decisions are formed locally at every node.

The above scheme should work well in a small overlay network with a handful number of nodes. However, in a large-scale overlay network, such as the Skype VoIP network mentioned below, link probing and information dissemination through flooding could incur significant overheads, since their costs grow quadratically as network size grows (measured in number of nodes). Moreover, if a node does not collect complete and up-to-date link information of the overlay network in time, it may not have sufficient information to choose better paths to other nodes not only for packets originating from itself but also for those to which it serves as a transit. In the worst case, the latter can be numerous if the node happens to lie on a "good" path as perceived by many other nodes with complete and up-to-date information. In this case,

the performance of an overlay network can be seriously degraded.

To improve scalability, reliability, and efficiency of overlay networks, we develop a low-cost end-to-end path probing and relay routing mechanism, called Path Probing Relay Routing (PPRR), which a node can deploy liberally for its own need. In the proposed approach, each node participating in the overlay network provides only one type of service: relaying packets for other participating nodes. Each node can then use this service to probe independently the quality of potential paths to its destination nodes, rather than running a full-fledged routing protocol. Furthermore, at any time it probes only a small set of paths, within which it then selects the best path(s) to convey the application traffic. It does not count on other nodes to exchange information on network link conditions obtained through probing elsewhere. Using simple heuristics, the proposed approach is capable of predicting path quality, selecting better alternative paths, and avoiding frequent path changes by using appropriate damping to reduce packet reordering which could affect the performance of some protocols such as TCP.

PPRR is of low cost for the following reasons. First, the probing is on-demand: only nodes that have packets to send will need to probe. Secondly, as we will see in Section IV, only a small number of paths need to be probed for the selection of alternative paths. Third, alternative paths need not use more than one relay hop.

For example, one of the potential applications that could benefit from PPRR is Skype, a popular, peer-to-peer VoIP system [7], or similar systems. The resulting improvements in reliability and latency of the underlying overlay transport network can help such a system provide better real-time voice services to its users.

The rest of this paper is organized as follows. In Section II, we review related works in the literature, comparing and contrasting several recently proposed approaches with ours. In Section III, we lay down our design goals, and, in Section IV, we present a detailed description of our top set system. In Section V, we report simulation results based on four datasets collected in real-world networks to evaluate the performance of the proposed approach. We discuss some of the potential issues in deploying the proposed approach in Section VI and conclude this work in Section VII.

II. Related Works

The Detour project [6] is aimed to solve a number of problems brought about by BGP as a consequence of its poor route selection. As the title of the project suggests, Detour attempts to achieve lower packet loss rates, smaller round-trip latency, and higher throughput by using “detouring” routes or relaying packets through intermediate nodes. They have focused on and have succeeded in collecting a large amount of real-world network statistics to show that BGP performs far from optimal: some paths incur a latency of 25% larger than the optimal, while many paths have a loss rate at least six times higher than their alternatives. They report that overall detouring can significantly improve about half of the paths. For the work of this paper, we were largely motivated by these statistics and evidence presented in [6]. However, we are not aware of any automated relay mechanisms similar to the system described in this paper that the Detour project has developed.

The Resilient Overlay Network (RON) project is aimed to provide optimized application-specific routing performance by means of rerouting packets in an overlay network [1][2]. Participating RON nodes constantly monitor the functioning and quality of virtual overlay links among themselves and flood the overlay network upon detecting link state changes. After obtaining the information of all virtual links, a RON node calculates the best route to all destinations using a link state routing algorithm. Using a fast detection algorithm, RON is able to recover from link failure within tens of seconds, compared with a few minutes achieved by BGP.

Our work departs from that of the RON project in several aspects. First, we employ an end-to-end path probing strategy that is more accurate and scalable than link probing. The probing overhead of a RON network grows quadratically as the size of the network grows, whereas in PPRR, it remains constant. Moreover, PPRR employs an on-demand probing technique in which nodes only probe when there is application traffic to transit. Secondly, as reported in RON, most of the alternative paths found by the link state algorithm involve only relaying through one intermediate relay node, so the search space is reasonably small, only growing linearly with the size of the network. This property enables us to use a simple randomized search strategy that searches for better alternative paths only among one-hop paths. Such an end-to-end probing together with the search strategy eliminates the need to flood the overlay network, as well as the problems of convergence and inconsistent views of the network topology that link state routing algorithms commonly face.

The Path Diversity with Forward Error Correction (PDF) system [4] optimizes delay-sensitive applications by means of spreading packets across multiple physically disjoint paths. It uses forward error correcting codes to encode packets, in an attempt to minimize extra bandwidth consumption. In order to obtain a set of disjoint paths, the system will invoke an all-pair TRACEROUTE session during the initial setup stage to collect topological information. We note that similar techniques can be applied to overlay network relay routing as well. Topological information obtained from TRACEROUTE can be used later on to form informed guesses as which potential relay nodes to consider during routing selection stage. Moreover, multiple paths can be used at the same time by relay routing with or without forward error correction applied to improve further end-to-end performance perceived by applications at the expense of extra bandwidth.

III. Design Goals

The primary goal of Path Probing Relay Routing (PPRR) of this paper is to alleviate occasional ineffectiveness and inflexibility of BGP to provide consistent high end-to-end performance to the applications. When the direct path reported by BGP experiences failure or the end-to-end performance of that path degrades to below a specific threshold, PPRR should find an alternative path of better performance and reroute application traffic using the latter path. In addition, PPRR should achieve this within a reasonably short amount of time,

so the applications can restore their normal end-to-end performance. For example, in many TCP implementations, a TCP connection for bulk data transfer will disconnect if either end does not receive any packet from the peer for several minutes. This means that any PPRR recovery time larger than several minutes is not acceptable from such an application's point of view.

In achieving its primary goal, PPRR should use resources sparingly to allow the scheme to be able to scale up. Rerouting traffic away from its shortest paths will result in an increased amount of total traffic on the network and, if not properly restrained, can lead to poor scalability, or even the collapse of the whole network, a phenomenon often cited as "the tragedy of the common." For example, a node can lower its own end-to-end packet loss rate by sending each packet multiple times or sending multiple copies of the packet through several paths; it achieves such a performance boost at cost of others. If many nodes do this, the overall performance of the network will surely degrade.

Finally, PPRR should allow incremental deployment as well as dynamic membership maintenance. Nodes should be able to join and leave the network at any time without resulting in massive service interruption.

IV. Description of The Top Set System

The PPRR system proposed in this paper uses an heuristic, called Top Set. The system consists of a set of autonomous nodes, operating in a decentralized, peer-to-peer manner. The nodes form an overlay network on top of the Internet, providing packet-rerouting services to its members. This rerouting service can be implemented using application layer tunneling and forwarding, or, alternatively, using network layer protocols like Source Demand Routing Protocol (SDRP) [3]. Based on such a rerouting service, the participating nodes decide their own probing and path selection policies and algorithms, independent of others.

At any time, each node maintains a set of viable routes, called the probe set, for each destination used by some active session at that node. The node will actively probe the viable routes and keep the probing results in a performance database. In normal situations when the direct path does not experience any performance problems, the node will send packets via the direct path. When the direct path experiences outage or performance degradation below a threshold, the node will select one of the viable routes to transport packets.

At the core of the scheme lie the path outage detection and the route selection algorithms. Because, as mentioned earlier, the optimal alternative path most often involves only one intermediate relay node, we need not run a full-fledged routing protocol at the overlay network level. Instead, we use an end-to-end path probing and learning mechanism, in which each node independently probes the quality of the potential one-relay paths to its destination node. For the simulation results reported in this paper, we use the average round-trip delay time as the main metric. We probe paths at randomized times to prevent overloading the network with synchronized probes. The obtained round-trip delay time samples are then fed into a weighted averaging scheme to calculate the path performance. The weighting constant will affect the response time of our scheme and should be determined empirically. The better half of the paths in the probe set, which we call the *top set*, are kept in the probe set, while the other half of the paths are replaced by a set of new paths, each going through a new randomly selected node.

Furthermore, when replacing a path, the target alternative path needs to be significantly better than the original path, e.g., by at least 5% better in terms of long-term average delay. This prevents route instability resulting from two or more paths that have comparable performance taking over one another at a high frequency. Even after a consistently better path has been found, the path being used to transport traffic will continue to be used for a grace period of time before path switching takes place, in order to reduce packet reordering that may degrade TCP performance.

We use a simple example, as is depicted in Figure 1, to illustrate the operation of the system described above. We assume that at the system-bootstrapping phase each node will get a list of all participating nodes in the overlay network. In addition, we suppose that there is an application on node S that has an active session with another application on node D.

In our scheme, probing and path replacements are conducted in a round-by-round basis, with each round lasting for a period of time, which, for this illustration, is one minute. We trace the operations of S from round 1 through round 39. Note that the scenarios presented below are arbitrary and are only for illustrative purposes. In round 1, node S randomly chooses a set of six nodes to be used as relay nodes. Based on these six nodes, S forms a probe set of six paths, each passing through one of the six nodes. (For notational simplicity, we call the path which is relayed through node X as path X.) S then starts probing these six paths using the randomized probing scheme described earlier. In round 2, roughly one minute later, S chooses paths A, B, and C to be included in the top set because they are the best three paths out of the six paths being probed in round 1. Meanwhile, S replaces the other three paths, paths D, E, and F, with three new paths, outside the probe set, which are relayed through three randomly chosen nodes G, H, and I. The same process continues for 19 additional rounds: S constantly replaces those paths that are not among the top performers with randomly chosen new paths.

In round 20, the direct path from S to D fails, and S detects such a failure after successive losses of probing packets in the

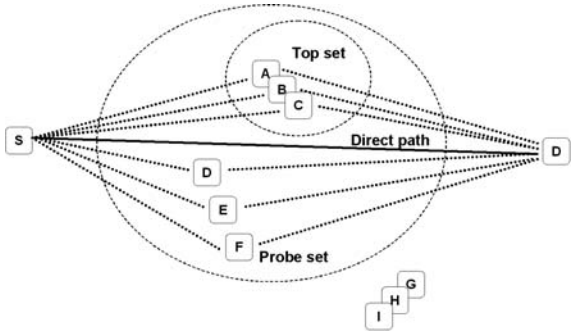


Figure 1. An example scenario to illustrate the operation of Path Probing Relay Routing (PPRR).

same round. In round 21, S reroutes all packets destined to D via path A, since at this moment this is the best alternative path available. In round 22, S finds by probing that path B performs 3% better than path A. However, the improvement is too insignificant (below a preconfigured threshold of 5%) to warrant a path replacement. Thus S continues using path A to transport its traffic to D. In round 24, path B starts to outperform path A by 7%, and the situation remains so until round 29, at which moment S switches to path B because it consistently outperforms path A by more than 5% for several consecutive rounds.

Finally, in round 35, the direct path has recovered from failure, so S switches back to the direct path in round 39 after observing an improvement of the direct path for several consecutive rounds.

V. Simulation Results

We use real-world traces obtained from various sources at different times to drive our simulation for more realistic results. Four datasets are used: RON1, RON2, PlanetLab1, and PlanetLab2. We first provide a description for each of them.

A. Description of Datasets

The RON1 and RON2 datasets were assembled by the RON project. RON1 contains 2,595,172 latency and loss samples, collected during Mar 21 to 23, 2001, on a network of 12 RON nodes, whereas RON2 contains 3,058,547 latency and loss samples, collected during May 7 to 11, 2001, on a network of 16 RON nodes.

The PlanetLab1 dataset contains all-pair 30-minute latency statistics on a network of 149 PlanetLab nodes, compiled by J. Stribling of MIT during Feb. 16 to Jun. 21, 2003. The statistics were collected by a central control node, which periodically polled the PlanetLab nodes to retrieve latency statistics obtained by the PING program in the past 30 minutes. In our simulation, we exclude those PlanetLab nodes that had too few statistics resulting from their occasional disconnection from the central node.

The PlanetLab2 dataset contains latency statistics gathered by the authors of this paper during Jun. 7 to Jun. 20, 2003, using a similar method as used in PlanetLab1 but with a finer granularity. Instead of 30 minutes, our central control node polled and retrieved statistics from the PlanetLab nodes every 5 minutes. Nodes with too few statistics are removed as well.

B. Four Approaches of Finding Alternative Paths

We report performance comparison results among four approaches of finding alternative paths. First, we refer to any full-fledged, link state based overlay routing algorithm as "*RON-like*," which requires nodes to flood the overlay network with link information before each node can conduct a local computation of new routes. Second, we refer to the approach that examines all paths passing through zero or one relay node and picks the best one among them as "*1-Relay*." Third, our proposed approach, as described in Section IV, is referred to as "*Top Set*," for it only maintains a small set of best paths discovered so far. For the simulation results reported in this section, we use a top set of three paths. We note that the probing cost of Top Set is much lower than that of 1-Relay, since the latter will need to probe all possible one-relay paths. Fourth, we refer to the approach that does not maintain any path performance statistics but picks a random node to relay traffic when direct path fails as "*Random*."

In our probing and path replacements algorithm, a round will last 1 minute for datasets RON1 and RON2, 30 minutes for PlanetLab1, and 5 minutes for PlanetLab2. These round times reflect the sampling frequency of the datasets. There are tradeoffs in determining the length of the round time: the shorter the round time is, the more responsive our algorithm becomes, but the cost of probing also increases. We have chosen these numbers partly because we are limited by the resolution of the data we obtained and partly as a result of our attempt to strike a balance between responsiveness and cost.

C. Average Length of Shortest Paths

We first analyze the datasets to compute the average length (in terms of number of relay nodes used) of shortest paths, using the average round-trip latency as the performance metric in path selection. For each dataset, we use a link state routing algorithm to compute shortest round-trip time for all source-destination pairs and record the number of relay nodes used in these paths. Figure 2 shows the cumulative density functions of the relay node counts of all shortest paths in the four datasets. It shows that there exist alternative paths that outperform the direct path (i.e., zero relay node used) for 46.1%, 42.4%, 60.7%, and 65.5% of the pairs in RON1, RON2, PlanetLab1, and PlanetLab2, respectively. (Note that we obtain these percentage numbers from Figure 2 by subtracting the probabilities at zero relay from 1.) In addition, direct paths and one-relay paths account for 94.7%, 94.2%, 77.2%, and 73.4% of the shortest paths in RON1, RON2, PlanetLab1, and PlanetLab2, respectively. Together, these results indicate that one-relay alternative paths are near optimal most of the time.

D. Availability and Predictability of Alternative Pathsl

In this section, we measure the *availability* and *predictability* of alternative paths when direct paths fail. For RON-like and 1-Relay approaches, we use complete information of the current network to compute the availability,

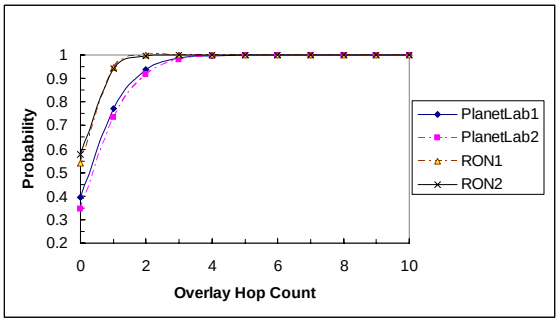


Figure 2. Cumulative density functions of the relay node counts of all shortest paths in the four datasets. These results show that the use of only one-relay paths can already achieve minimum latency with a probability of at least 70%. (Note that curves for PlanetLab1 and PlanetLab2 almost completely overlap, and so do those for RON1 and RON2.)

which can be viewed as the upper bounds for the availability that we can successfully find an alternative path when the direct path fails. The availability for the Top Set approach is the probability that there is a better alternative path among the set of current top paths. The availability for the Random approach is the probability that the source node can reach the destination node using a randomly picked relay node.

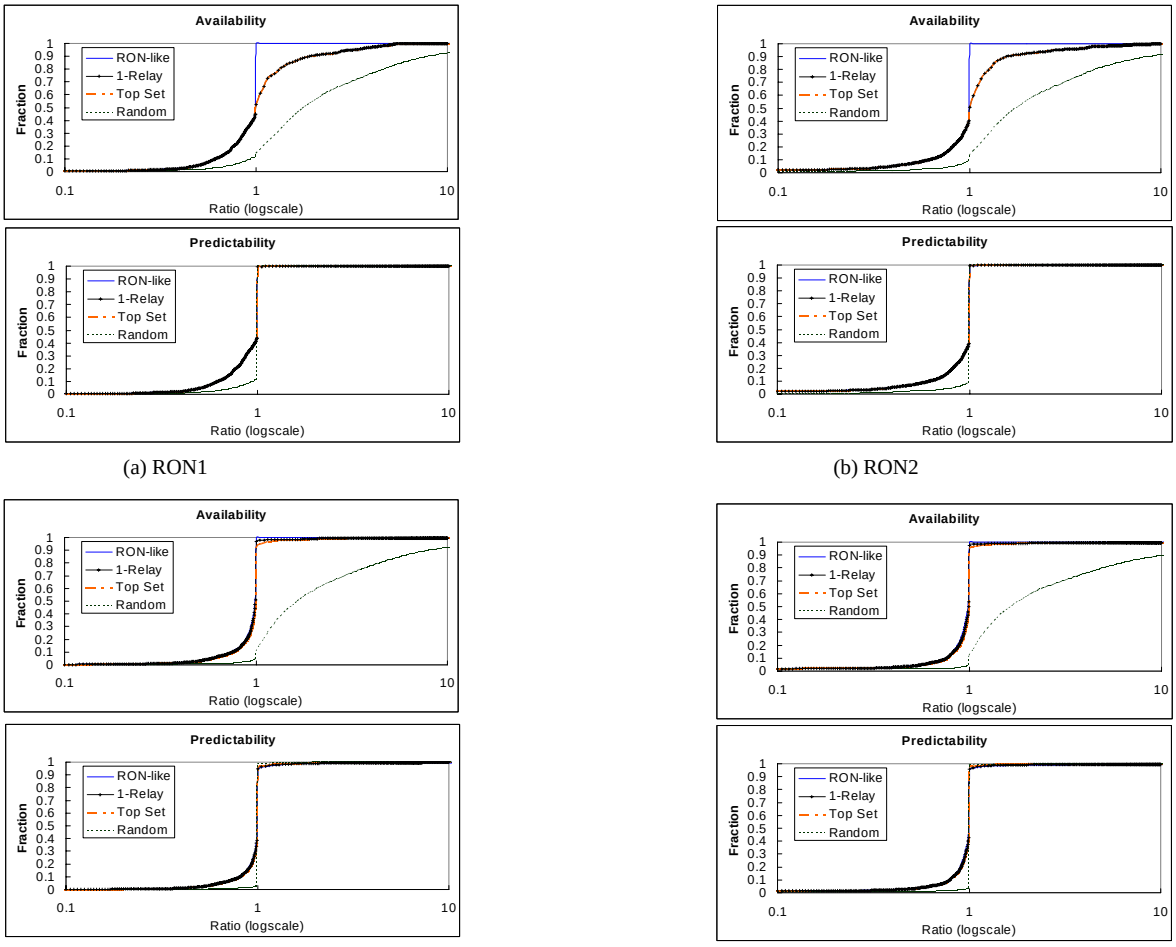
The predictability of an approach, on the other hand, means the probability that the best path discovered before the current round is still a better alternative path when the direct path fails. Note that after probing, an approach may select the direct path as the best path for the next round.

As shown in Table I, 1-Relay performs almost the same as RON-like. The proposed Top Set approach performs well, with performance very close to RON-like and 1-Relay, and is much better than Random. Note that all the approaches exhibit their worst performance for the PlanetLab1 dataset due to its low sampling frequency (once every 30 minutes). Moreover, we note that Random performs poorly for the RON2 dataset. This is because there are fewer "better alternative paths" available in RON2, and consequently it is difficult for the Random approach to find a good alternative path.

E. Relative Performance of Alternative Paths

We compare the quality of the alternative paths selected by the four approaches with that of the direct path. Figure 3 shows the ratio of average round-trip latency of selected alternative paths to that of direct paths, where the average is taken over all source-destination pairs at all rounds. In order to show the quality of alternative paths, direct paths are *not* used by 1-Relay, Top Set, and Random; only RON-like is allowed to use direct paths. Thus the reported ratios represent upper bounds for the first three approaches.

The top portions of the four data sets results assume that precise information for route calculation is available. The results show that the performance of Top Set is very close to that of 1-Relay, while the performance of the Random approach lags far behind. In addition, 22.5%, 18.6%, 10.6%, and 11.9% of the alternative paths selected by RON-like, 1-Relay, and Top Set improve by 20% or more over direct paths



(a) RON1 (b) RON2

(c) PlanetLab1 (d) PlanetLab2

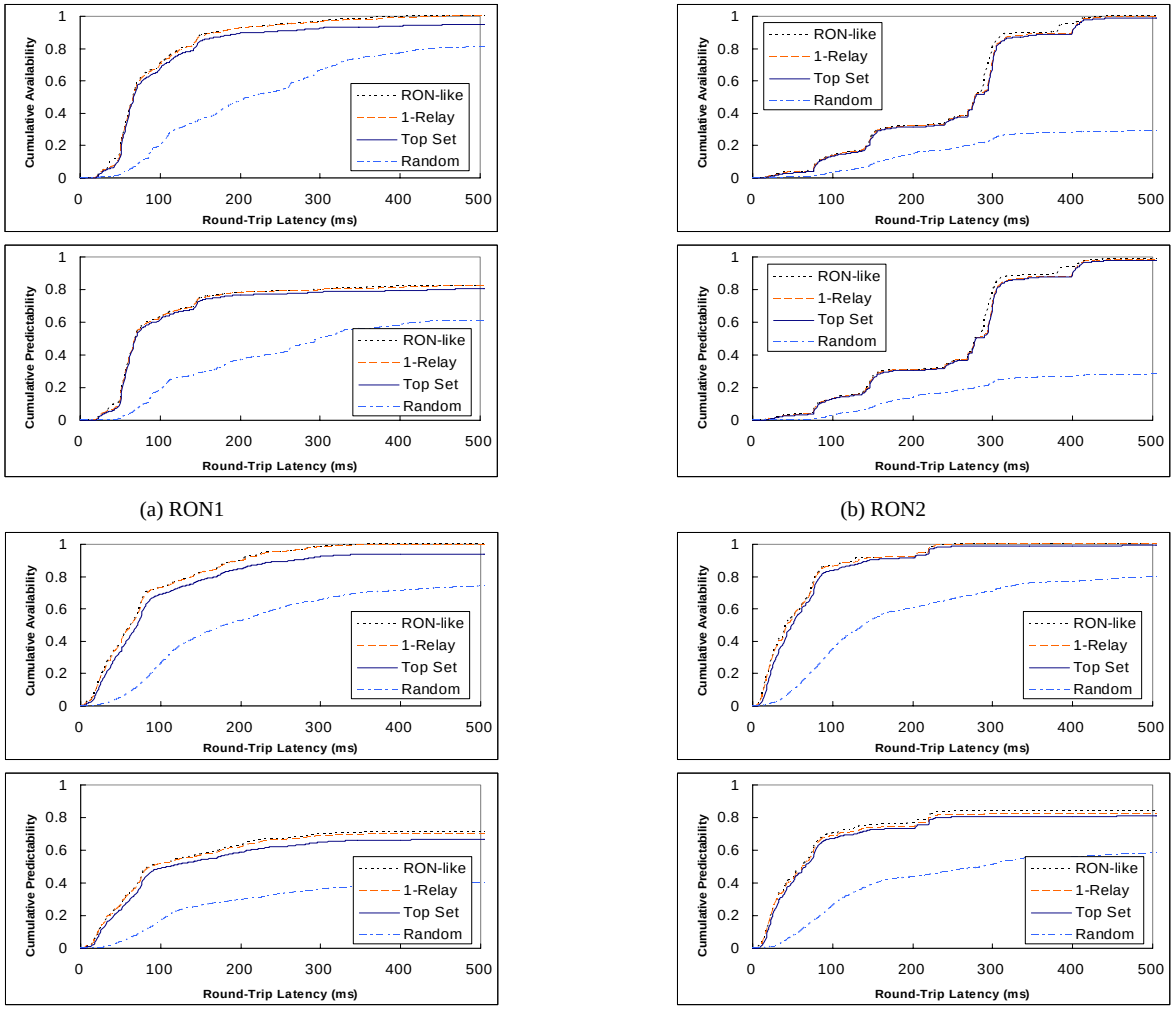
Figure 3. Performance of alternative paths for (a) RON1, (b) RON2, (c) PlanetLab1 and (d) PlanetLab2; top of each sub-figure: cumulative density of alternative path round-trip latency (relative to direct path) using precise information for route calculation; bottom of each sub-figure: cumulative density of alternative path round-trip latency (relative to direct path) using information collected in prior rounds for route calculation. Note that the performance of alternative paths found by RON-like, 1-Relay, and Top Set are very close, so their curves almost overlap.

in RON1, RON2, PlanetLab1, and PlanetLab2, respectively. However, 25.3%, 23.3%, 2.9% and, 2.2% of the alternative paths selected by 1-Relay and Top Set are impaired by 20% or more compared to direct paths for RON1, RON2, PlanetLab1, and PlanetLab2, respectively. This is because direct paths perform very well in some situations. This means that we should stick to direct paths if no significantly better alternative paths are found.

For this reason, we use a simple prediction strategy, in computing the bottom portions of the four data set results. That is, we will switch to an alternative path only if it outperforms the direct path by at least 5% for three consecutive prior rounds; otherwise we remain using the direct path. However, if the direct path is found to be disconnected (e.g., all PING messages fail), then the switching to an alternative path will take place immediately after the current round. The bottom portions of Figure 3 report round-trip latency ratios for the four approaches when such a simple prediction method is used. Due to inevitable prediction errors, the improvement rates are slightly lower than those reported earlier in this section (22.2%, 18.5%, 8.6%, and 9.8% for RON1, RON2, PlanetLab1 and, PlanetLab2, respectively). But the impaired rates are significantly improved (0.1%, 0.2%, 1.8% and, 1.1% for RON1, RON2, PlanetLab1, and PlanetLab2, respectively) because the direct paths will still be used when there are no significantly better alternative paths.

F. Absoluate Performance of Alternative Paths

In this section we report in Figure 4 the average round-trip latency of the best available and the alternative paths found by the four approaches (under our simple prediction strategy). By comparing the results, we note that the performance of alternative paths found by 1-Relay is as good as that found by RON-like, and that our Top Set approach performs very close to RON-like and 1-Relay.



(a) RON1

(b) RON2

(c) PlanetLab1

(d) PlanetLab2

Figure 4. (a) RON1, (b) RON2, (c) PlanetLab1 and (d) PlanetLab2 performances in face of direct path failure; top: cumulative density of alternative path round-trip latency using precise information for route calculation; bottom: cumulative density of alternative path round-trip latency using information collected from prior rounds for route calculation. In all cases, the performance of the top set approach is comparable to that of RON-like. Note that the performance of 1-Relay and that of Top Set are almost indistinguishable, which is evident from the two overlapping curves representing 1-Relay and Top Set.

TABLE I. PROBABILITY OF AVAILABILITY AND PREDICTABILITY OF ALTERNATIVE PATHS FOR FOUR APPROACHES WHEN DIRECT PATHS FAIL. NOTE THAT, IN ALL CASES, THE PERFORMANCE OF THE TOP SET APPROACH IS COMPARABLE TO THAT OF RON-LIKE.

Dataset	%	RON-like	1-Relay	Top Set	Random
RON1	Avail	100	100	95.1	81.3
	Pred	82.4	82.4	80.3	60.9
RON2	Avail	100	99.4	98.9	29.8
	Pred	99	98.3	97.9	28.6
PlanetLab1	Avail	100	99.7	94.2	75.8
	Pred	71.4	70.8	66.7	40.7
PlanetLab2	Avail	100	100	99.3	81.8
	Pred	84.1	82.3	81.0	59.4

G. *Impact of Number of Top Paths and New Paths on Performance of Alternative Path*

As discussed in Section B, the availability of one-relay alternative paths roughly remains constant as network size grows. This suggests that the probing overhead of PPRR can be kept constant if all we need to find is a viable alternative path when the direct path fails. We can increase probing to improve performance if, in addition to availability, we want to find a better alternative path.

There are tradeoffs between the number of paths to be kept in the top set and the number of new paths to be added in each round in order to achieve higher performance. In the experiments so far, we keep 2 top paths and pick 2 new paths for RON1 and RON2 datasets, and 3 top paths and 3 new paths for PlanetLab1 and PlanetLab2. The total number of the paths to be kept and picked in the experiments is near lg N, where N is the total number of nodes. In general, the more top and new paths to be included, the better performance may be achieved, but this will incur more cost. Figure 5 shows the effects of the probe set size and number of new paths on the predictability. Since four datasets have similar results, we show only the results of PlanetLab1, where 149 nodes are measured. It shows that as the numbers of top and new paths increase, the predictability increases dramatically at the beginning but insignificantly after a critical threshold.

VI. DISCUSSION

Like the loose source route option, PPRR can create an opportunity for security breaches under IP address based packet authentication. When the direct path fails, packets will be routed through an intermediate relay node on an alternative path and hence will bear the relay node's IP address as the source address when they reach the destination. Now a compromised PPRR node can spoof the source IP address of a packet by falsely claiming that it is relaying the packet for the source. Hence, IP address based packet authentication schemes can not be used with PPRR; the users need to adopt end-to-end cryptographic protocols for authentication purposes.

There are also economical concerns: there appears to be disincentives for a node to relay packets for other nodes, for it can consume extra bandwidth and degrade the nodes' capacity to serve normal traffic. Like all peer-to-peer systems, PPRR will need incentive stimulating mechanisms to prevent participating nodes from deliberately delaying other nodes' packets, or even refusing to relay packets.

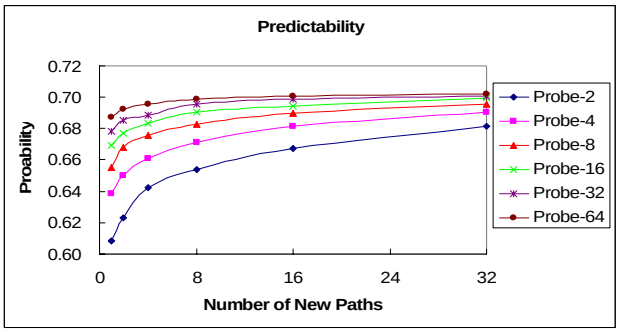


Figure 5. Effects of the probe set size on predictability of alternative paths when direct paths fail. The results show that a probe set size of 16 can already achieve near-optimum. performance.

Although the probing cost of PPRR remains constant as network size grows, it does increase linearly with the number of active source-destination pairs in the network. In a busy network, however, there might be many source-destination pairs, and the probing cost of PPRR may become significant. We note that it is possible to aggregate probing traffic when multiple paths join at a particular overlay link to eliminate redundant probes. Also, it is possible to prioritize traffic such that only destinations of important traffic get to use the PPRR service.

VII. CONCLUSION

In this paper, we propose Path Probing Relay Routing (PPRR) for overlay networks. The scheme is scalable, as the probing overhead is independent of network size. We report trace-based simulation results based on real-world traffic traces, showing that the performance of PPRR is comparable to that of a full-fledged, link state routing algorithm. Compared to the latter, PPRR uses less probing bandwidth and eliminates the need to flood the overlay network with link state information. Furthermore, the simulation results demonstrate that simple and low-cost heuristics such as Top Set suffice to find good relay paths.

REFERENCES

[1] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris, "The case for resilient overlay network," in *Proceedings of HotOS VIII*, May 2001.

[2] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris, "Resilient overlay networks," in Proceedings of Symposium on Operating Systems Principles, 2001.

[3] D. Estrin, T. Li, Y. Rekhter, K. Varadhan, and D. Zappala, "Source demand routing: packet format and forwarding specification (version 1)," RFC 1940, Internet Engineering Task Force, May 1996.

[4] T. Nguyen and A. Zakhor, "Path diversity with forward error correction (PDF) system for packet switched networks," in *Proceedings of IEEE INFOCOM*, 2003.

[5] Y. Rekhter and T. Li, "A border gateway protocol 4 (BGP-4)," RFC 1771, Internet Engineering Task Force, March 1995.

[6] S. Savage, T. Anderson, A. Aggarwal, D. Becker, N. Cardwell, A. Collins, E. Hoffman, J. Snell, A. Vahdat, G. Voelker, and J. Zahorjan, "Detour: a case for informed Internet routing and transport," *IEEE Micro*, vol. 19, no. 1, pp. 50-59, January 1999.

[7] "P2P Telephony Explained," http://skype.com/skype_p2pexplained.html