RC24059 (W0609-106) September 25, 2006
Computer Science

IBM Research Report

How to Select a Good Alternate Path in Large Peer-to-Peer Systems?

Teng Fei[1], Shu Tao[2], Lixin Gao[1], Roch Guerin[3]

[1]Department of Electronic and Computer Engineering
University of Massachusetts
Amherst, MA

[2]IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

[3]Department of Electronic and Systems Engineering
University of Pennsylvania

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

How to Select a Good Alternate Path in Large Peer-to-Peer Systems?

Teng Fei†, Shu Tao*, Lixin Gao†, Roch Guérin*
† Dept. Elec. & Comput. Eng., U. Massachusetts, Amherst
Email: {tfei, lgao}@ecs.umass.edu
* Dept. Elec. & Sys. Eng., U. Pennsylvania
Email: shutao@seas.upenn.edu, guerin@ee.upenn.edu

***Abstract*— When multiple paths are available between communicating hosts, application quality can be improved by switching among them to always use the best one. The key to such an approach is the availability of diverse paths, i.e., paths with uncorrelated performance. A promising approach for implementing the necessary path diversity is to leverage the capabilities of peer-to-peer systems. Peer-to-peer systems are attractive not only because their many participating nodes can act as relays for others, and therefore offer a large number of different alternate paths, but also because their distributed operation can facilitate the deployment of the required functionality. However, these advantages come at a cost, as the sheer number of alternate path choices they offer creates its own challenge. In particular, because not all choices are equally good, it is necessary to develop mechanisms for easily and rapidly identifying relay nodes that yield good alternate paths. This paper is about the formulation and evaluation of such mechanisms in the context of large peer-to-peer systems. Our goal is to devise techniques that for any given destination allow nodes to quickly select a candidate relay node with as small a cost as possible in terms of how much information they need to store or process. We combine several heuristics that rely only on local routing information, and validate the resulting solution by comparing it to a number of benchmark alternatives. This comparison is carried out using both topology data from RouteView/RIPE and PlanetLab nodes, and through measurements across a large set of PlanetLab nodes.**

I. Introduction

Recent research as well as commercial offerings have demonstrated the potential of path switching as a means for improving end-to-end application performance [1], [2], [3], [4]. Path switching assumes the availability of multiple paths between communicating hosts, and adaptively chooses the path that provides the best performance. Although the current IP routing infrastructure does not intrinsically support multi-path routing, the diverse paths required by path switching can be obtained through end system-based solutions. A particularly promising approach is to use peer-to-peer overlay networks, which typically require little infrastructure support from the underlying network, and are therefore relatively easy to deploy. Peer-to-peer systems often include a large number of geographically distributed nodes through which a rich set of alternate overlay paths can be constructed. Furthermore, a peer-to-peer architecture makes it easier to ensure that most of the participating nodes can provide the functionality needed to act as relay points on behalf of other nodes. Overlay networks are a viable option to provide alternate paths because the increasing availability of broadband Internet access means that performance bottlenecks are often not the access links, but instead intra-AS links within backbones or inter-AS links [5], [1], [3]. This makes overlay routing an effective method to bypass performance degradations on Internet paths, and peer-to-peer system an ideal vehicle for offering the path diversity needed for path switching.

Specifically, we consider applications with reasonably stringent performance requirements, e.g., voice or video, and implemented in the form of a large peer-to-peer system, e.g., a "Skype-like" environment [6], [7]. Communications between peers can take place not only over the default path as determined by IP routing, but can also rely on an alternate overlay path through a relay node selected among other peers. Alternate paths are limited to two-hop overlay paths, since a two-hop overlay path has been shown [8] to be typically sufficient to bypass most performance degradations[1] on the default path. Communication is switched to the overlay path during periods of poor performance on the default path, and path switching decisions must incur minimum latency to accommodate the stringent nature of the application performance requirements. This means that an alternate path needs to be identified, and possibly initialized, at the time of session initiation. Furthermore, while it is possible to simultaneously maintain several alternate paths, this would result in additional overhead. Therefore, it is desirable to identify *ahead of time* one relay node that offers a suitable alternate path. The focus of this paper is in identifying and evaluating a simple yet efficient rule that can be used to pre-select a relay node, when initiating communication between two nodes in a large peer-to-peer based system.

As mentioned above, the choice of a peer-to-peer environment is motivated by its ability to provide a large and diverse set of alternate paths. However, the sheer number of choices it offers creates problems of its own, when it comes to pre-selecting a single relay node that can provide a good alternate path for reaching a particular destination. Intuitively, a good alternate path should exhibit end-to-end performance characteristics that are both reasonably good and not correlated with those of the default path. The first challenge is that we need to deal with a large number of

[1] We define performance more precisely later in the paper.

possible choices whose "goodness" depends on the target destination. Secondly, which relay node offers a good alternate path is also likely to vary over time. As a result, a brute-force solution that continuously monitors all possible alternatives is neither feasible nor desirable. Conversely, randomly selecting a relay node, while clearly lightweight, can often result in poor choices. A natural approach is then to seek overlay paths that share the minimum amount of physical resources with the default path, under the assumption that disjoint paths are also likely to exhibit uncorrelated performance.

The approach proposed in this paper is based on this intuition and attempts to identify overlay paths that are as disjoint as possible from the default path at the *AS level*. We focus on AS-level path disjointness, as opposed to physical or IP-level disjointness, because this greatly reduces the amount of topology information that needs to be maintained by end-systems. However, even keeping track of AS-level paths between every pair of nodes[2] in a large peer-to-peer system can be a challenging task. It is, therefore, necessary to further reduce the amount of information needed by a node to effectively select alternate paths. For that purpose, we introduce a heuristic, the *earliest-divergence* rule, that achieves a reasonable trade-off between the amount of information that needs to be maintained and processed and a node's ability to easily identify good alternate paths. The rule relies only on "local" AS-level information and favors relay nodes whose own AS-level path deviates from the default path at the earliest possible point. As we expand in Section III, this is clearly an approximation, but as we shall demonstrate it performs reasonably well when it comes to helping select good alternate paths.

Another contribution of our work is to develop methodologies to evaluate the performance of our approach across a reasonably broad range of configurations. We take advantage of the Oregon RouteViews [9] and RIPE [10] routing data and the PlanetLab testbed [11] to validate our approach from different perspectives. We not only investigate the robustness of our approach in finding disjoint paths, but also verify its effectiveness in finding paths that offer uncorrelated end-to-end performance. We developed and deployed distributed software on the PlanetLab testbed to measure the performance correlations between all possible overlay paths and the default path. This allows us to evaluate our approach against other schemes (e.g., random selection, best selection, etc.). We also investigate the robustness of our path selection scheme across different network performance metrics, such as delay degradation and loss. Our study reveals that the selection of best overlay path is relatively constant across different performance metrics, which means that (i) we do not need different rules for selecting relay nodes as the target performance metric varies, and (ii) we can evaluate our approach using the performance metric (i.e., delay degradation) that is the easiest to monitor given the scale of our measurements.

[2]This information is needed in order to assemble a complete, end-to-end overlay path between a source and a destination node through an arbitrary relay node.

The remainder of this paper is organized as follows. Section II reviews existing works related to our study. Section III motivates and introduces the heuristics we designed for alternate path selection in a peer-to-peer environment. Section IV demonstrates the effectiveness of our approach using topology data obtained from both the RouteView and RIPE databases and the PlanetLab testbed. Section V further validates the proposed approach using measurement traces collected on the PlanetLab testbed. Finally, Section VI summarizes our findings and points to a few possible extensions.

II. Related Work

Several recent studies have demonstrated the effectiveness of using multiple paths to improve application performance [8], [12], [13], [3], [4]. For instance, the RON project [8] uses alternate overlay paths to bypass path failures. Tapestry [12] exploits overlay path redundancy in a structured peer-to-peer system. However, these works do not fully address the problem of how to select backup paths whose performance exhibits as little correlation as possible with that of the default path. As a result, the backup path may experience poor performance at the same time as the default path, and therefore not provide the best possible alternative for avoiding end-to-end performance degradations.

To address this issue, overlay path selection algorithms that take advantage of topology information have been proposed in several settings [14], [15], [16]. The approach of [14] assumes that the joint overlay link failure probabilities are known. However, obtaining these probabilities in a decentralized peer-to-peer system can consume significant resources. The solution proposed in [15] relies on end-to-end probing of the overlay paths and the inference of the loss probabilities on the underlying physical path segments, which obviously has scalability limitations. Similarly, the authors in [17] use `traceroute` to obtain the IP level path as well as the latency information, to estimate the path disjointness between the direct path and the overlay paths, and then choose the k most disjoint paths using a disjointness threshold. As pointed out by the authors, such a method is only suitable for small-scale overlay networks, and may need to resort to other more scalable approaches as the number of nodes in the overlay network grows.

In order to limit resource requirement, more recent studies have focused on reducing the end-to-end measurement needed to select overlay paths. In [16], the authors propose a routing underlay dedicated to topology probing. With the help of this underlay, one can use inferred AS path information to construct disjoint paths between communicating nodes. The potential problem of this method is in the accuracy of AS path inference. For instance, [18] showed that AS paths inference can often be much less accurate than expected. In [19], Gummadi *et al.* select relay nodes by randomly choosing k overlay nodes and picking the one with the best performance. Clearly, when k is large such a system performs well if monitoring a large number of paths is feasible. When k is small, on the other hand, random selection can potentially eliminate good relay nodes. In particular, in our setting we

consider the case where $k = 1$, as initializing the probing of $k > 1$ paths at the time of a path switching decision in order to pick the best one, would represent too much of an overhead and typically incur a latency that would not be compatible with the requirements of real-time applications.

Our approach differs, therefore, from earlier works in that we take advantage of measurement information in selecting one "best" relay node, while keeping the resources required for making this decision to a minimum. In particular, we rely only on information that is local to the nodes responsible for making the selection. Furthermore, our focus is on applications with relatively stringent performance requirements, such as video or voice, for which an alternate path must be kept readily available to take over in case of performance degradation on the direct path. Our heuristics for selecting good relay nodes rely only on AS path information available at the source node, where it can be acquired using light-weight measurements such as `traceroute` and `ping`. As a result, the amount of storage and processing required at each node is minimized. As we show in the paper, the use of local measurements in guiding relay node selection is essential in ensuring good alternate paths in many cases. In particular, while a simple random selection is sufficient in some cases, there are many instances where the number of good alternate paths is small compared to the total number of potential candidates, so that a blind, random selection would be unlikely to yield an acceptable choice.

III. Alternative path selection: A heuristic approach

A. Disjoint overlay path

Ideally, the paths used for path switching should have end-to-end performance that exhibits as little correlation as possible (negative correlation would be even better). In particular, the paths should not experience performance degradations simultaneously, so that if performance degrades on one path, traffic can be switched onto another and vice versa. More precisely, let X_1 denote the event that path 1 experiences poor performance, and X_2 the event that path 2 experiences poor performance. Under an ideal path switching scenario, the overall probability of performance degradation is

$$P_{degrad} = P\{X_1 X_2\} = P\{X_2\} P\{X_1 | X_2\} \quad (1)$$

Given a default Internet path, our goal is to find alternate paths that minimize $P\{X_1 X_2\}$. Therefore, we require the alternate path to not only have relatively good performance (i.e., minimize $P\{X_2\}$), but also have performance that has as little correlation as possible with that of the default path (i.e., minimize $P\{X_1|X_2\}$). A direct way of identifying good alternate paths is to monitor the performance variations on all possible candidates simultaneously. The path that minimizes $P\{X_1 X_2\}$ is then considered to be the best choice. However, this is not a scalable solution, especially when the number of available relay nodes (hence the number of available candidate overlay paths) is large.

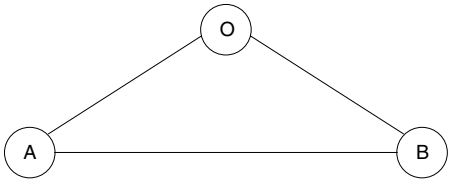


Fig. 1. An example of disjoint overlay path and default path.

Another approach is to seek paths that exhibit as little overlap as possible with the default path. The assumption is that the fewer physical resources shared by two paths, the less likely their performance will be correlated. Clearly, this assumption does not always hold, particularly when performance degradations are not contributed equally by all links. However, as we shall see in our measurement study, although there are almost certainly other factors besides overlapping links that can affect performance correlation between paths, using path disjointness as a selection criterion does yield paths with significantly lower performance correlation.

Consider the example of Fig. 1, where the default path between two nodes A and B is denoted by $P_{AB} = (x_0, x_1, ..., x_n)$, with x_i representing the different segments (e.g., IP hops or ASes) on the path. The alternate overlay path consists of two "hops": the first hop is the path from node A to relay node O, denoted by $P_{AO} = (y_0, y_1, ..., y_m)$; the second hop is the path from node O to node B, denoted by $P_{OB} = (z_0, z_1, ..., z_l)$. We define S as the set of segments on P_{AB} that overlap with segments on either P_{AO} or P_{OB}. Thus, finding an alternate path with the least overlap with the default path amounts to selecting a relay node O among all possible relay nodes, so that $|S|$ is minimized.

B. Disjoint paths: AS-level or IP-level?

An important aspect when attempting to identify the most disjoint alternate path is the definition of a "node" on the path, as it affects $|S|$ and, therefore, the measure of path disjointness. One option is to examine paths at the lowest possible granularity, i.e., at the IP-level, where nodes are routers. Although such a choice seems to be natural at first, it suffers from several problems. First, the `traceroute` routine, the most commonly-used method for retrieving IP-level path information, relies on ICMP messages that are often either ignored, or rate-limited by routers [20]. As a result, reliably and accurately obtaining IP-level path information can be challenging for many source-destination pairs. This is compounded by the fact that both IGP and BGP routing changes affect IP-level paths. In addition, as we shall see later, accurately predicting the end-to-end disjointness of IP-level paths based on only *local* information, i.e., the results of `traceroute` from the source node only, is also difficult.

In this paper, we consider path disjointness at the AS-level. The main method for identifying the set of ASes crossed en route to a destination is again `traceroute`. There has been several studies [20], [21] on how to convert IP-level paths

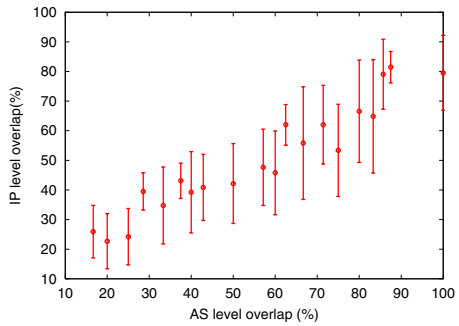


Fig. 2. A comparison between AS-level and IP-level path disjointness.

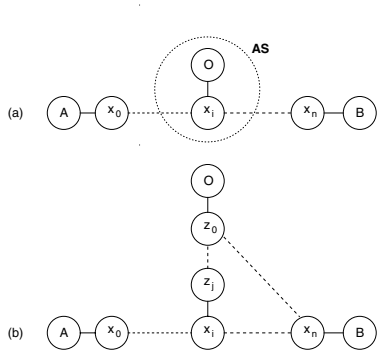


Fig. 3. An example of applying the earliest-divergence criteria based on (a) IP-level path information and (b) AS-level path information.

into AS-level paths. Furthermore and most importantly, even if `traceroute` returns incomplete IP-level path information, it is still possible to accurately infer the ASes that nodes with unknown IP addresses are associated with. Therefore, focusing on AS-level path information has the advantage of greater (AS-level) accuracy and lower overhead. The disadvantage is that the most disjoint path at the AS-level are not necessarily the most "physically" disjoint, which may influence performance correlation. In particular, ASes come in many different sizes and the number of IP hops traversed in different ASes can vary significantly.

In Fig. 2, we compare the overlap between the direct path and all possible overlay paths connecting 42 PlanetLab nodes (associated with 40 different ASes). For each path pair, we compute the percentage of overlapping nodes with respect to the total number of nodes on the direct path, using both IP-level and AS-level path information. The plot shows the mean and standard deviation of the IP-level overlap as a function of the AS-level overlap. We see that although the two are not in perfect agreement, they are sufficiently correlated that an AS-level predictor can provide a reasonably accurate estimate of IP-level overlap between paths.

C. The earliest-divergence rule

Assuming that AS-level path information is available between all pairs of nodes in the system, it is straightforward to find the overlay path that is most disjoint from the default path. However, when the number of nodes is large, as in a large peer-to-peer system, maintaining even AS-level path information for all node pairs becomes difficult. In a system with N nodes, the number of communicating node-pairs is $O(N^2)$, and the total number of potential relay nodes can be as high as $N-2$. Allowing every node to identify the most disjoint AS-level path to all possible destinations calls for each node to perform $O(N^2)$ path comparisons. It is, therefore, important to devise a method for identifying maximally (or nearly maximally) disjoint paths with a lower overhead. In this paper, we propose to use the *earliest-divergence* rule for that purpose.

The earliest-divergence rule is an approximation to the maximum disjointness criteria. It has the advantage of relying only on local information available at a node. Specifically, candidate relay nodes, and hence alternate paths, are selected based on only the AS-level path information from the source node to them. The rule prefers paths that diverge from the default path at the earliest possible point. If in Fig. 1, we use $P_{AB} \wedge P_{AO}$ to denote the number of AS hops at which P_{AO} diverges from P_{AB}, the earliest-divergence rule selects candidate relay nodes such that $P_{AB} \wedge P_{AO}$ is minimized. The intuition behind this approximation is that if the AS path to a node diverges early from the default path, the AS path from that node tends to merge back into the default path relatively late. In particular, since AS paths follow specific patterns determined by AS relationships [22], it is unlikely that an overlay path diverging from the direct path at a certain inter-AS link will also merge back into that path via the same inter-AS link. As illustrated next, this intuition is, however, less applicable when considering paths at the IP-level.

Consider the configuration shown in Fig. 3(a). O is a possible relay node located in an AS that happens to also be present in the direct path P_{AB}, $P_{AB} = (x_0, x_1, ..., x_i, ..., x_n)$, $P_{AO} = (x_0, x_1, ..., x_i)$, and $P_{OB} = (x_i, x_{i+1}, ..., x_n)$. Here, $x_0, x_1 ..., x_n$ represent IP-level nodes along the path. If the earliest IP hop at which any overlay path diverges from P_{AB} is x_i, O will then be selected by the earliest-divergence rule. However, since in this case link x_i is an intra-AS link, path P_{OB} merges right back into path P_{AB}, so that the resulting overlay path P_{AOB} fully overlaps with the direct path P_{AB}. This is an extreme example that illustrates that applying the earliest-divergence rule at the IP-level can result in very poor choices, i.e., complete overlap between the direct and overlay paths. Similar even if less extreme outcomes can be obtained even if node O is not directly connected to an intra-AS link x_i on the direct path. In particular, whenever node O is in an AS that belongs to the direct path, path P_{OB} will very often merge back into P_{AB} within the same AS as nodes O and x_i belong to. In all such cases, early divergence at the IP-level yields paths that are hardly disjoint from the direct path.

In contrast, applying the earliest-divergence rule at the AS-level can significantly increase the likelihood of obtaining disjoint paths. This is because it guarantees inter-domain divergence between the overlay path and the direct path. As shown in [22], [23], AS paths typically follow a "valley-free"

pattern. Therefore, once an overlay path diverges from the direct path to another AS, the chance that it will merge back via the same AS is small. For example, consider node O in Fig. 3(b), which is a relay node such that P_{AO} diverges the earliest from P_{AB} at the AS level at link x_iz_j (note that $x_0, x_1, ..., x_n$ and $z_0, z_1, ..., z_j$ now all represent ASes, and nodes A, O, and B are associated with ASes x_0, z_0, and x_n, respectively). Link x_iz_j can be one of three types of inter-AS links: peering, customer-provider, or provider-customer [22]. If x_iz_j is a peering link between ASes x_i and z_j, P_{OB} will merge back into P_{AB} (hence result in a full overlap between paths P_{AOB} and P_{AB}), if and only if P_{OB} contains exactly an "up-hill" segment $(z_0, .., z_j)$ (i.e., links in this segment are all customer-provider links), the peering link z_jx_i, and a "down-hill" segment $(x_i, ..., x_n)$ (i.e., links in this segment are all provider-customer links). Similarly, if x_iz_j is a provider-customer or customer-provider link, full overlap occurs if and only if path P_{OB} contains exactly the segment $z_0, ..., z_j$, link z_jx_i, and the segment $(x_i, ..., x_n)$, and also follows a "valley-free" pattern according to the relationships and routing policies of each individual AS [22]. These are relatively stringent conditions, which greatly increase the likelihood that the overlay path selected by the earliest-divergence rule remains disjoint from the default path at the AS-level (hence at the IP-level as well).

In the next few sections, we demonstrate that the earliest-divergence rule is indeed successful in both reducing the number of candidate relay nodes that a node can choose from, and increasing the likelihood of finding a (maximally) disjoint path among the remaining nodes. Nevertheless, there are cases where the number of remaining candidates remains large even after applying this rule, and more importantly where these remaining candidates still include a non-negligible fraction of relatively poor choices. It is, therefore, necessary to extend the earliest-divergence rule to not only further reduce the number of candidate nodes it produces, but to do so while eliminating those associated with alternate paths that exhibit limited disjointness with the default path.

D. Selecting the best candidates

We extend the earliest-divergence rule by selecting relay nodes that are "far" from the direct path. The intuition is that by going far away from the default path, the likelihood that the overlay path quickly merges back into the direct path is reduced. In order to implement this intuition, we assume that the source node A knows the round-trip delay from itself to the subset of relay nodes selected by the earliest-divergence rule (denoted as D_{AO}). For example, A can send `ping` probes to relay nodes to obtain this information. Similarly, the source node also knows the round-trip delay between itself and the destination node (denoted as D_{AB}).

We further assume that the application using the default path has a maximum acceptable delay, e.g., a real-time application such as VoIP, so that imposing an upper bound on the round-trip delay of alternate paths is also necessary. We denote this upper bound as β. Our modification of the earliest-divergence rule then proceeds as follows: Among the set of relay nodes produced by the earliest-divergence rule, we first eliminate those that might yield a round-trip delay greater than β. Since the source node A is unaware of the round-trip delay between overlay node O and destination node B, we use $D_{AO} + D_{AB}$ as a worst case estimate for D_{OB}. Thus, we select nodes that satisfy

$$D_{AOB} = D_{AO} + D_{OB} \leq 2D_{AO} + D_{AB} \leq \beta. \quad (2)$$

Second, the remaining nodes are sorted in decreasing order of their delay D_{AO}, and only the first m (m is typically less than 10) are kept. The relay node that will be used to construct an alternate path is randomly chosen among these m remaining candidates.

To evaluate the effectiveness of this extended earliest-divergence heuristic, we need to compare the set of paths it produces with the results of several other benchmarks. First, we verify that a naïve solution that randomly selects an alternate path from all possible candidates often results in a bad choice, while our heuristic yields significantly better choices. In addition, we also compare our scheme with another random scheme, namely, the random-k scheme of [19], which incurs a greater overhead in either the number of paths that need to be set up or in the latency in finding a good alternate when path switching needs to take place. Second, we compare the paths selected by our heuristic to the best possible choices. This is necessary to assess the penalty we have incurred because of the several approximations on which the heuristic relies, i.e., path disjointness as an approximation for performance independence between two paths, AS-level disjointness as an approximation for IP-level disjointness, and the earliest-divergence criterion with delay extension as an approximation for selecting disjoint overlay paths based on full path information.

The next two sections explore the above issues. Section IV focuses on establishing that the extended earliest-divergence heuristic is capable of producing reasonably small candidate sets that include a majority of maximally or near-maximally disjoint paths. Section V extends the investigation to establish that the resulting disjoint paths make for suitable alternates when used with path switching to avoid performance degradations. We also demonstrate that the results hold across a broad range of performance degradations, as defined by various delay and loss metrics.

IV. Topology-based study

In this section, we evaluate the effectiveness of the earliest divergence rule in finding disjoint overlay path. Our evaluation is based on AS-level topology data.

A. Topology data set

In order to evaluate AS-level path disjointness after applying the earliest-divergence rule, we use two data sets that provide us with AS-level path information. The first data set is obtained from the routing tables archived on May 10, 2005 by the Route Views project at the University of Oregon and the

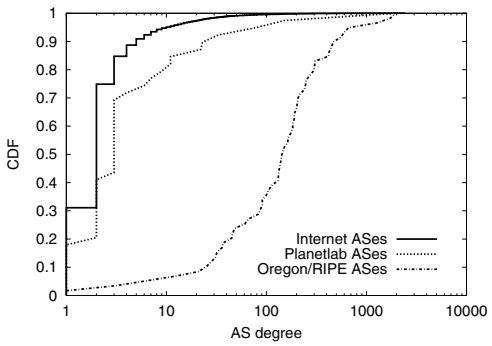


Fig. 4. AS degree distributions of the Planet-lab ASes, the Oregon/RIPE ASes, and 19,932 ASes in the Internet.

RIPE network coordination center. We denote this data set as Oregon/RIPE. We use this data set to emulate a peer-to-peer overlay network, with the peers of the Oregon/RIPE BGP observer as the overlay nodes of the peer-to-peer network. The AS-level path information among those peering points are construct using the BGP routing information, derived using the longest prefix match rule. Note that although most peering points are from top tier ASes, these peering points do not necessarily export all routes to Route Servers. Therefore, the AS paths between some of the BGP peers are not directly observable from the BGP routing tables. In the following analysis, we use 56 BGP peering points of Oregon/RIPE for which we can construct an AS-level path between each pair of points to create a full mesh.

Our second data set consists of AS-level paths obtained from the PlanetLab testbed. These paths were obtained by performing `traceroute` measurements between nodes in the PlanetLab testbed. The PlanetLab testbed contains more than 500 machines at about 250 different locations. We chose 41 nodes and performed pair-wise `traceroute` measurements. The resulting IP-level routes were then mapped into AS-level paths.

TABLE I

SUMMARY OF TWO DATA SETS

Data set	Nodes	ASes	Average AS degree	Src-dst pairs
Oregon/RIPE	56	56	262	3080
PlanetLab	41	39	61	1640

Table I shows a summary of the two data sets. Although the size of the two data sets are somewhat similar, they have different characteristics. Nodes from the Oregon/RIPE data set are mostly from major network providers that have rich connectivity. In contrast, most of the PlanetLab nodes are hosted by universities or research institutes that typically belong to networks with lower connectivity than that of nodes in the Oregon/RIPE data set. Fig. 4 shows the CDF of the degrees of the 56 ASes from the Oregon/RIPE data set and the 39 ASes from PlanetLab nodes. For comparison, we also show the degree distribution of 19,932 ASes that appear in the Oregon/RIPE data set. We can see that the degrees of the Oregon/RIPE ASes are much higher than those of the PlanetLab ASes. Furthermore, the degree distribution of PlanetLab ASes is closer to that of the Internet ASes in general. This difference in the two data sets allows us to evaluate the earliest-divergence rule in different environments, in which nodes in a peer-to-peers system are connected via different types of networks.

B. Effectiveness of the earliest-divergence rule

1) Reducing the number of candidate relay nodes: We first investigate the effectiveness of the earliest-divergence rule (indicated as ED) in reducing the number of candidate relay nodes. Figs. 5 (a) and (b) show the CDF of the number of candidate relay nodes before and after applying the rule for all source-destination pairs in the Oregon/RIPE and PlanetLab data sets, respectively. We observe that the earliest-divergence rule reduces the number of candidates noticeably for most source-destination pairs. However, the improvements are somewhat different for the two data sets. The number of relay nodes is reduced more significantly for paths in the PlanetLab data set. For instance, for about 50% of source-destination pairs, the number of remaining relay nodes is reduced from 39 to less than 5. However, the number of relay nodes is only slightly reduced for about 50% of the source-destination pairs in the Oregon/RIPE data set. The discrepancy comes from the fact that the degree of Oregon/RIPE ASes is typically higher than that of the PlanetLab ASes. Therefore, AS paths in Oregon/RIPE data set usually have many alternate paths that diverge at the second hop. As a result, most of the relay nodes remain after applying the earliest-divergence rule.

TABLE II

PERCENTAGE OF SOURCE-DESTINATION PAIRS WHOSE BEST OVERLAY PATH HAS n-HOP OVERLAP WITH THE DIRECT PATH, BEFORE AND AFTER APPLYING THE EARLIEST-DIVERGENCE RULE

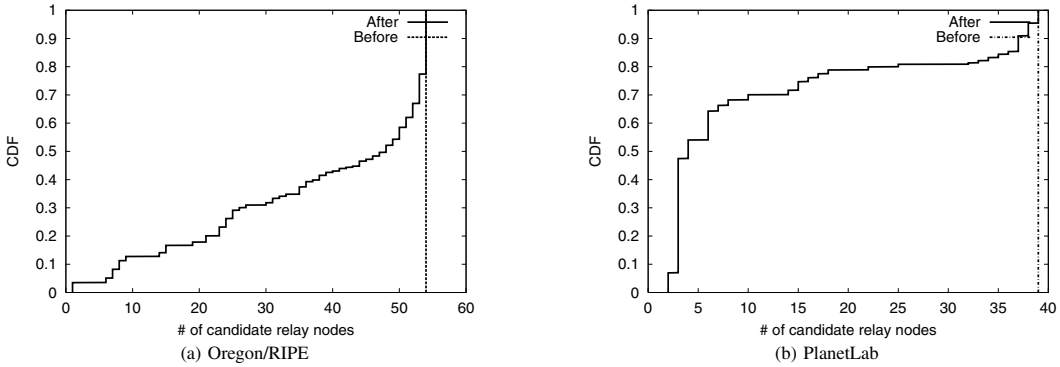
Overlap hop count (n)	0	1	2	3	4	5	6
Oregon/RIPE (before)	92.8	3.7	3.5	0.0	0.0	0.0	0.0
Oregon/RIPE (after)	92.8	2.6	3.4	1.1	0.1	0.0	0.0
PlanetLab (before)	79.5	19.6	0.9	0.0	0.0	0.0	0.0
PlanetLab (after)	77.4	19.5	1.0	0.1	1.2	0.7	0.0

TABLE III

PERCENTAGE OF SOURCE-DESTINATION PAIRS WHOSE WORST OVERLAY PATH HAS n-HOP OVERLAP WITH THE DIRECT PATH, BEFORE AND AFTER APPLYING THE EARLIEST-DIVERGENCE RULE

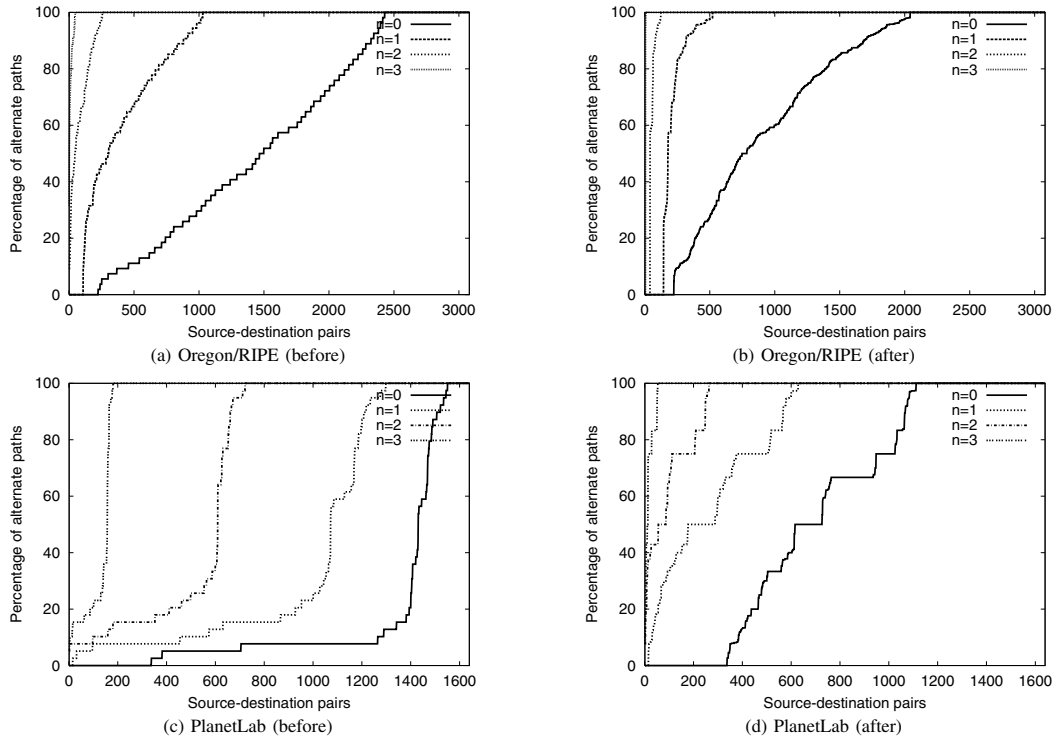
Overlap hop count (n)	0	1	2	3	4	5	6
Oregon/RIPE (before)	21.2	45.4	25.1	6.7	1.3	0.2	0.0
Oregon/RIPE (after)	33.7	49.4	12.9	3.7	0.2	0.0	0.0
PlanetLab (before)	5.5	15.4	35.1	32.6	7.8	3.2	0.4
PlanetLab (after)	32.3	29.4	21.6	11.6	2.9	2.0	0.2

2) Finding disjoint overlay paths: We further evaluate the extent to which the earliest-divergence rule helps us find overlay paths that are most disjoint from the direct path. For that purpose, we count the number of overlapping ASes between the direct path and the overlay paths excluding the source and destination ASes. That is, 0-hop overlap means that the direct and overlay paths share only the source and destination ASes; 1-hop overlap means that the direct and



(a) Oregon/RIPE (b) PlanetLab

Fig. 5. Number of candidate relay nodes before and applying the *earliest-divergence* rule.



(a) Oregon/RIPE (before) (b) Oregon/RIPE (after)

(c) PlanetLab (before) (d) PlanetLab (after)

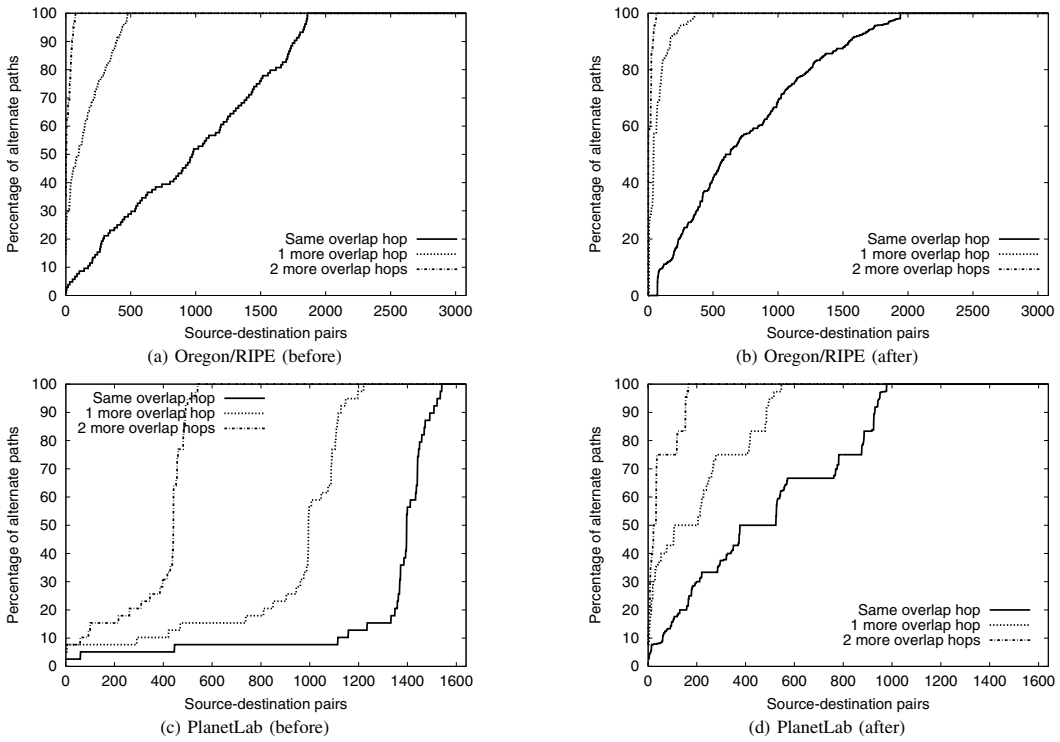
Fig. 6. Distribution of node pairs whose best overlay path overlaps with the direct path at $\leq n$ hops, before and after applying the *earliest-divergence* rule.

overlay paths share one AS in addition to the source and destination ASes, and so on.

We first look at the earliest-divergence rule's performance, in terms of retaining those "good" relay nodes that have small overlapping with the direct path, and rejecting those "bad" relay nodes that have large overlapping with the direct path. Table II shows the percentages of source-destination pairs whose best overlay path has an n-hop overlap with the direct path, before and after applying the rule. We see that the earliest-divergence rule is capable of retaining the best relay nodes for most source-destination pairs. We then examine its ability to reject bad relay nodes. Table III shows the percentages of source-destination pairs whose worst overlay path has an n-hop overlap with the direct path, before and after applying the earliest-divergence rule. It can be observed that this rule also does a reasonably good job in removing bad overlay paths, especially when their overlap with the direct path is ≥ 2 hops.

Next, we show that the earliest-divergence rule can improve the likelihood of selecting a good overlay node. In Fig. 6, we plot the percentages of relay nodes that can produce an overlay path that overlaps with the direct path for n or less hops, before and after applying the earliest-divergence rule. As can be seen from the figure, overlay paths that have significant overlap with the direct path are more likely to have been discarded after applying the rule. This further confirms our intuition that an overlay path that diverges early from the direct path is also likely to be more disjoint from the direct path.

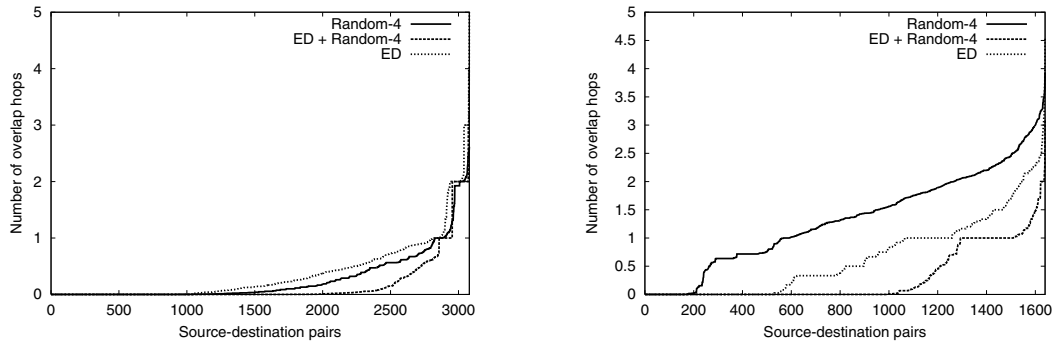
Thirdly, we show that the earliest-divergence rule can improve the odds of selecting a path whose overlap with the



(a) Oregon/RIPE (before) (b) Oregon/RIPE (after)

(c) PlanetLab (before) (d) PlanetLab (after)

Fig. 7. Percentage of overlay paths that achieve the same number of overlapping ASes or 1 or 2 hops more than the best overlay path (i.e., the most disjoint from the direct path), before and after applying the earliest-divergence rule.



(a) Oregon/RIPE (b) PlanetLab

Fig. 8. Distribution of the number of overlapping hops between direct paths and overlay paths using random-4 and earliest divergence (ED), and the combination of the two rules.

direct path is close to that of the best possible overlay path (i.e., the most disjoint from the direct path). Fig. 7 plots the likelihood that the selected relay path has either the same number of overlapping hops as or 1 or 2 hops more than that of the most disjoint overlay path, before and after applying the earliest-divergence rule. We see that in general, applying the earliest-divergence rule increases the likelihood of selecting relay nodes that can offer an overlay path with a similar number of overlapping hops compared to the best achievable.

Finally, we compare the earliest divergence rule with the *random-k* relay selection scheme proposed in [19]. The random-k scheme selects k relay nodes randomly from all the candidate relay nodes. It then sends probing packets to monitor the performance of the k paths and selects as the overlay path the path whose response packet is first returned. In [19], the authors also evaluate the performance of the scheme for various values of k and find out that a value of $k = 4$ appears to achieve a reasonably good compromise between being able to quickly identify a good alterante overlay path and the resulting probing overhead. In our analysis, we therefore compare the earliest divergence rule with a *random-4* rule, which randomly selects 4 relay nodes from the entire relay node pool, and then chooses the path with the least overlap among the 4 paths. We then take the average performance of all possible 4-node combinations for each source-destination pair, and compare it to that achieved by the earliest-divergence rule. In addition, we also apply the random-4 scheme to the subset of candidate relay nodes that are pre-selected by the

earliest-divergence rule, as opposed to randomly choosing them among all possible candidate nodes. The results are shown in Figure 8.

We can see that for the data set of Oregon/RIPE, the earliest-divergence and the random-4 rules perform similarly for about 1/3rd of the source-destination pairs. This is because for ASes that are well connected, a random selection is likely to yield overlay nodes that diverge early from the direct path. For other source-destination pairs, the random-4 rule only slightly outperforms the earliest-divergence rule. Note that we are comparing the "best" or the least overlapping path selected by the random-4 rule with the average performance of the earliest divergence rule.

For the Planet-lab data set, however, the earliest-divergence rule outperforms random-4 for all source-destination pairs. These results suggest that, for networks that are well connected, the random-k rule is likely to generate similar or even better results compared to the earliest-divergence rule, while for networks that are not well connected, the random-k rule may not perform as well. This is because in such configurations, the end networks have few alternate paths that diverge early from or converge late back on the direct path. As a result, a random selection rule, such as the random-k rule, is less likely to select reasonably disjoint paths, while the criterion on which the earliest-divergence rule is predicated allows it to more consistently select disjoint paths. This can also be verified by comparing the random-4 rule with the "earliest-divergence + random-4" rule, where the more discriminate selection criterion of the earliest-divergence rule allows the random-4 rule to now select paths from a set of more *disjoint* relay paths than the original set. As can be seen, the "earliest-divergence + random-4" outperforms the base "random-4" for most source-destination pairs, especially in the PlanetLab data set.

Based on the above analysis, we conclude that in most cases the earliest-divergence rule is reasonably successful in selecting overlay paths that are disjoint from the direct path, while significantly reducing the total number of candidate relays. We also note that when the source node belongs to a well-connected ISP network, the earliest-divergence rule is less successful in significantly reducing the number of candidates. In such cases, the delay constraint extension proposed in Section III-D will prove helpful.

V. Measurement-based study

A. Measurement methodology

Although the results from the previous section helped validate the earliest-divergence rule's ability to produce paths that are disjoint at the AS-level, they did not validate it in terms of performance. In order to carry out such a validation, we used the PlanetLab testbed to conduct measurements to determine and compare the end-to-end performance correlation between the default path and various overlay paths. We developed and installed a distributed measurement program on all accessible PlanetLab nodes. Using this program, we can simultaneously monitor the end-to-end performance of paths between multiple source and destination nodes. For purposes of illustration, consider Fig. 9 where nodes A and B are the target source and destination, with multiple overlay paths $P_{AO_1B}, P_{AO_2B}, ..., P_{AO_nB}$ available to supplement the default path P_{AB}. We initiate measurements on all paths from node A. For the direct path P_{AB}, periodic UDP probes are sent from node A to node B, and an ACK packet is sent back on path P_{BA} for each received probe. For an overlay path P_{AO_iB}, probes are sent first from node A to node O_i, then forwarded by node O_i to node B. ACK packets are sent in the reverse direction, i.e., from node B to node O_i, then to node A. By analyzing the packet traces recorded for each path, we can assess the level of correlation that exists between the paths, and in particular which overlay path offers the least correlated end-to-end performance with the direct path. Because of the relatively high overhead introduced by the measurements, we mainly study paths originating from nodes at UPenn and UMN. Since many PlanetLab nodes are connected to educational networks, the default paths from UPenn and UMN to these nodes often go through the Abilene network, which is likely to bias the measurement results. To avoid this problem, we enforce a special source-based routing policy at the gateway routers at both universities, so that the measurement traffic is always routed onto the commercial provider networks (Cogent for UPenn, Wiltel for UMN).

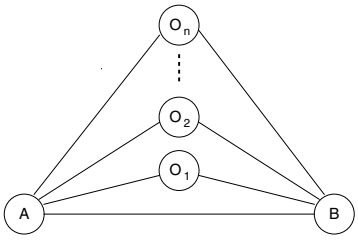


Fig. 9. Measurement setup on the PlanetLab testbed.

B. Performance correlation metrics

Different applications often have different requirements in terms of performance. Consequently, we need to investigate different metrics when evaluating performance correlation between paths. We focus our study on end-to-end delay variations and losses, since they are the main performance parameters that affect application quality.

Focusing first on delay variations as an indicator for performance degradations on a path, we denote the round-trip delay measured on path i as D_i. The random variable D_i has a mean $\overline{D_i}$ and a standard deviation σ_i. When a probe samples the round-trip delay on path i, we consider that the path is experiencing a delay degradation at the time of sampling if

$$D_i > \overline{D_i} + k\sigma_i. \quad (3)$$

Here k is a constant and can be adjusted to define different levels of delay degradations. In particular, if the delay on an alternate path j satisfies

$$D_j \leq \overline{D_j} + k\sigma_i, \quad (4)$$

we consider that path j is not experiencing the same level of delay degradation as path i, therefore is a good alternate for path i. As we shall see later, which paths are good alternates to avoid delay degradations is relatively insensitive to the exact definition of delay degradation. With the above definition, we can measure the percentage of delay degradations on the default path that can be avoided by using an alternate path, i.e., the joint probability $P\{X_1X_2\}$ scaled by $P\{X_1\}$ in Eq. (1). This provides a metric for comparing the suitability of different alternate paths in supplementing the default path. More generally, we define delay degradation using a moving window as follows. With the definitions of Eqs. (3) and (4), we consider a path to be experiencing delay degradations in the current time window, if the percentage of delay degradations is greater than a threshold α within a window of size L. Otherwise, performance is considered to be good. We then compute the percentage of delay degradations on the default path that can be avoided by using different alternate paths, and use this metric for measuring path correlation in terms of delay degradations.

We define a similar metric for loss performance. For a series of packet loss samples, we use a moving window to measure the average loss rate over a certain time period. Given a window size L, we define a path to be experiencing a loss period, if its average loss rate in the current window is higher than a threshold γ. Meanwhile, if the loss rate on path j is less than γ, we consider path j to be a good candidate for avoiding loss performance degradations on path i. The loss avoidance percentage is again used to capture the suitability of different alternate paths.

We first study whether the "goodness" of an alternate path is affected by the specific performance metric one considers, i.e., delay or loss degradation. We measure the performance of 11 paths connecting two nodes at UPenn and UMN and two nodes at UPenn and UMass, respectively. The paths consist of one default Internet path and overlay paths using 10 different PlanetLab nodes as relay nodes. The measurements were conducted over a full 48-hour period. During the measurements, probes were sent simultaneously on all paths every 20 ms. We pick all possible combinations of path pairs from among the 11 paths, and select one as the "primary" path and the other as the "alternate" path. For each primary-alternate path pair, we compare the delay and loss avoidance percentages. In this comparison, we compute these percentages using a moving window of size $L = 500$. For delay degradation, we use a threshold of $\alpha = 10\%$, and a constant $k = 3$. For loss degradation, the threshold was set to $\gamma = 5\%$. As mentioned below, different combinations were examined and the results were found to exhibit relatively little sensitivity to the specific values chosen.

Fig. 10 shows the results of this comparison in the form of a scatter plot. As can be seen from the figure, path pairs with high delay avoidance percentage also tend to have high loss avoidance percentage, and similarly when the delay avoidance percentage is low, so is it for loss. This suggests that if two paths are uncorrelated delay-wise, they are also likely

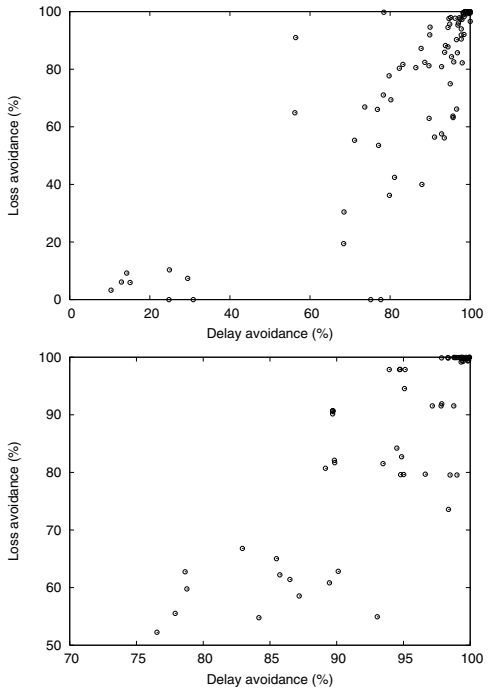


Fig. 10. A comparison between the percentages of loss degradation avoidance and delay degradation avoidance for different path pairs between UPenn and UMN (top), and between UPenn and UMass (bottom).

to be uncorrelated loss-wise, and vice versa. We revisited this finding for other values of k ($k = 1, 2, 4, 5, 6$) and γ ($\gamma = 1\%, 3\%, 10\%$), and these observations remained true in spite of some variations. We therefore conclude that performance correlation between two paths can be evaluated using either delay or loss as the metric of choice. In the remainder of the paper, we use mainly delay degradation as our performance metric, as compared to the relatively rare loss events, delay variations are easier to measure. In particular, our measurements require monitoring numerous paths between a large set of source and destination node pairs. Using delay as our metric greatly simplifies the trace collection task.

C. Performance evaluation

Our heuristic is predicated on the assumption that AS-level disjoint paths are likely to have uncorrelated performance. In this section, we verify this assumption by evaluating the performance of paths selected using our approach. The evaluation is based on the measurement methodology described in Section V-A.

In our measurement, we study two sets of paths, originating from the nodes at UPenn and at UMN, respectively. We selected 47 PlanetLab nodes to be either the destination node for these paths or a candidate relay point for an overlay path. For each source-destination node pair, we simultaneously measure the round-trip delay on the default path, as well as on all possible overlay paths (46 in total). Probes are sent

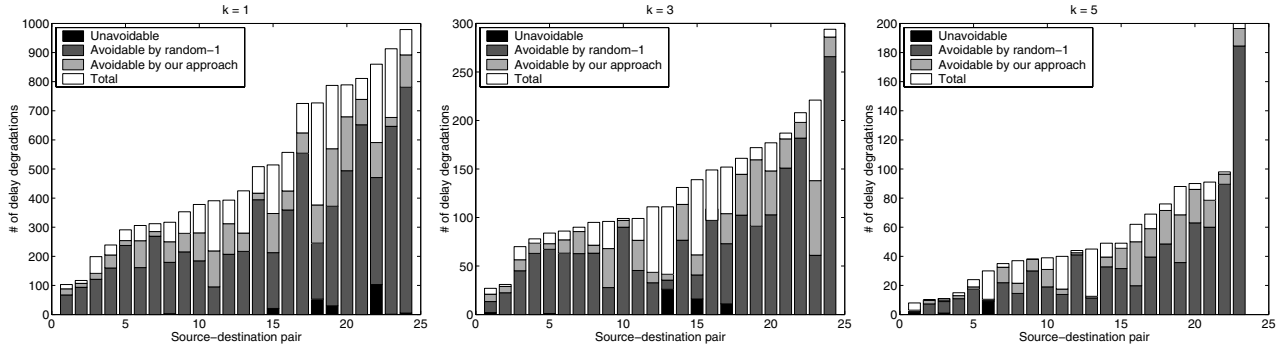


Fig. 11. The numbers of total delay degradations, unavoidable delay degradations, delay degradations avoidable by randomly selecting an alternate path, and delay degradations avoided using alternate paths selected by our approach. The definition of delay degradation is varied by configuring $k = 1, 3, 5$, respectively.

every 500 ms[3], with measurements lasting 1 hour for each source-destination pair. Because of the previously mentioned greater ease in generating accurate measurement results, we use delay degradation as the metric to evaluate performance across paths. From the traces we collect, we can find the number of delay degradations on the default path, as well as the number of degradations that can be avoided by using a particular overlay path. Fig. 11 plots these numbers ($k = 1, 3, 5$) for 30 direct paths originating from UPenn[4]. For each direct path, we plot four different parameters: total number of degradations, number of unavoidable degradations (i.e., degradations that occurred on all paths and therefore can not be avoided no matter which alternate path we use), average number of degradations avoidable by randomly selecting an alternate path, and average number of degradations avoidable using the paths selected by our approach (i.e., using the earliest-divergence rule with delay constraint, as defined in Eq. (2).

From the figure, we can make the following observations. First, a few delay degradations are unavoidable. Unavoidable degradations are most likely introduced by the access links, which are shared by all paths, overlay and default. However, the number of unavoidable degradations is relatively small. As a result, path switching can be quite effective in most cases in improving end-to-end performance. Second, in many cases a randomly selected alternate path only avoids a small fraction of delay degradations. Most likely because performance on the selected path and the default path are correlated. The heuristic we have developed allows us to minimize this correlation. This can be seen from the improvement that results from using the paths selected by the heuristic. The results are qualitatively similar across all values of k, even if the absolute numbers of delay degradations observed obviously vary as a function of k.

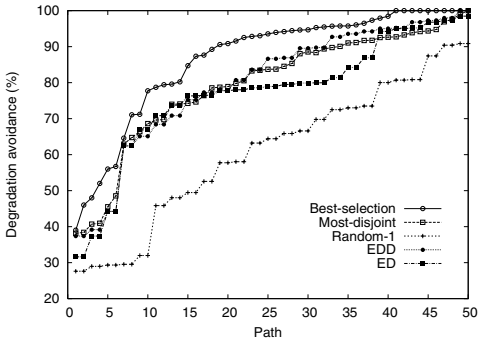


Fig. 12. A comparison between schemes in their ability to pick good alternate paths.

Fig. 12 provides yet another perspective on the performance of the proposed heuristic. It compares the percentage of avoided delay degradations across five different path selection schemes. It uses nodes at either UPenn or UMN as sources, and 30 PlanetLab nodes as destinations. 46 other PlanetLab nodes served as possible relay points. The heuristic is first compared to (1) *random-1*, i.e., randomly selecting one alternate path out of the 46, (2) *best-selection*, i.e., selecting the best alternate path based on an off-line analysis of all the measurements, and (3) *most-disjoint*, i.e., selecting the path that is the most disjoint from the direct path at the AS level. We first apply the earliest-divergence heuristic alone. As shown in the figure, this significantly improves the overall percentage of delay degradation avoidance when compared to the random-1 scheme. However, the number of choices in the reduced path set is still quite large (20.4 nodes when averaged over all source-destination pairs). We then extend the heuristic by applying the previously mentioned delay constraint (indicated as *earliest-divergence with delay* or EDD). We use a delay bound of s = 400 ms (a typical value to ensure end-to-end VoIP quality) and select relay nodes randomly from the three nodes that have the largest delay from the source after

[3]The choice of a value of 500 ms, instead of the previously mentioned value of 20 ms used for exploring the correlation between loss and delay degradations, is primarily due to the larger number of paths we monitor simultaneously. The 500 ms value ensures that end-system performance does not become an issue.

[4]The results do not include all 47 direct paths that were being monitored, because failures or disruptions on a number of PlanetLab nodes resulted in incomplete or inaccurate measurements for some nodes. Furthermore, the number of paths also varies when we use different definitions of delay degradation (e.g., if k is large, no degradation is observed on some paths. These paths will be excluded from our analysis).

applying the delay constraint (i.e., $m = 3$). This further improves the results and yields significantly higher degradation avoidance percentages. In general, our approach does not yield performance that is as good as the best path selection. This is mainly because the assumption that the most AS-disjoint paths have the most uncorrelated performance is not always valid. This can be demonstrated by the performance gap that exists between the best-selection and the most-disjoint schemes. However, our scheme is able to achieve a performance as good as the most-disjoint path selection scheme. Because of the delay constraint applied, the average performance of our scheme is even better in some cases, as also shown in Fig. 12.

Last, we compare our scheme to the more general *random-k* scheme studied in [19]. As discussed earlier, unlike the random-k scheme, which randomly selects k alternate paths and uses the best, our scheme makes biased selections and favors alternate paths that are more disjoint from the direct path. Intuitively, given a certain value of k, the random-k scheme is more effective if the set of candidate paths includes a large proportion of good choices. However, if this proportion is small (i.e., as in cases where *random-1* performs poorly), our approach is more likely to outperform random-k. To confirm this, we compare the percentage of delay degradation avoidance using both our scheme and the *random-4* scheme for 25 paths originating from a node at UPenn. We sort the results in ascending order of the performance difference between our scheme and random-1. Specifically, for the path labelled 1, the initial density of good choices is high, so that the random-1 rule is as good or even better than the EDD rule. Conversely, for path number 25 the initial density of good choices is low so that a random selection is unlikely to yield a good outcome, while the EDD rule successfully prunes the set of candidates to significantly increase the odds of making a good choice. As shown in Fig. 13, as the path number increases, i.e., the initial candidate set includes fewer and fewer good choices, the EDD rule typically outperforms the random-4 scheme by a growing margin. Furthermore, if we first apply the earliest-divergence (ED) rule to the entire candidate set and then use the random-4 selection on the resulting subset of nodes, the performance of the random-4 scheme also improves significantly. This is because the random-4 scheme now samples from a smaller set of candidates with a higher density of good choices. This confirms the observations of Section IV that our scheme can not only outperform random-k in many cases, but also help improve its performance by improving the set of nodes it chooses from. However, note that as mentioned earlier, the environment we consider calls for the *a priori* selection of *one* alternate path that is to serve as a "hot standby" for the direct path in case of performance degradation. This is different from the implicit goal of the random-k rule. Pre-establishing k alternate paths represents too much of an overhead, and waiting for a response to probe packets after experiencing performance degradations on the direct path is likely to introduce too much latency for the type of real-time applications we consider. As a result, the random-k rule may not be appropriate in such a setting.

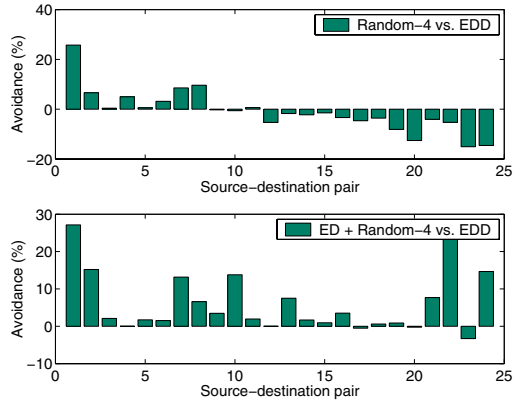


Fig. 13. (a) Performance difference between the random-4 scheme proposed in [19] and the earliest-divergence with delay constraints (EDD) scheme (top); (b) performance difference between the combined scheme of earliest-divergence (ED) and random-4, and EDD (bottom).

TABLE IV

THE INFORMATION ABOUT THE 9 PLANETLAB NODES

Node	Domain name	AS path from source
1	planet3.berkeley.intel-research.net	55:16631:7018:15861:7018
2	planetlab1.dtc.umn.edu	55:16631:3356:217
3	planetlab-1.cs.princeton.edu	55:16631:4969:1785:88
4	pli2-br-1.hpl.hp.com	55:16631:3356:8553:1889
5	planetlab1.comet.columnbia.edu	55:16631:1:6395:20274:14
6	planetlab1.nbgisp.com	55:16631:3356:18473
7	planetlab1.cs.ubc.ca	55:16631:6327:271:852
8	planetlab4.millennium.berkeley.edu	55:16631:2152:25
9	planetlab1.singapore.equinix.planet-lab.org	55:16631:4637:4657:9989

The above measurement results have shown the average performance of our approach for alternate path selection. Due to the large scale of the measurements and the associated overhead, we evaluated performance correlation between paths using delay degradation as our metric. In the next section, we study a specific example in which we focus on a smaller set of paths and monitor their performance using more frequent probes and over a longer period of time. This allows us to perform a loss-based study and provide an additional perspective to assess the effectiveness of the proposed approach.

D. A case study for path selection

In this example, the source and destination nodes are at UPenn and UMN, respectively. The AS path between the two nodes is 55:16631:3356:57. In order to better demonstrate the potential of our approach, we choose 9 PlanetLab nodes as potential relays through which to build alternate paths. Information about these 9 nodes is given in Table IV. We probe the default path as well as the 9 overlay paths simultaneously using a probing interval of 20 ms. We monitor the performance of all paths for 48 hours, and use the recorded traces to analyze the resulting performance, assuming that path switching is used between the direct path and a selected overlay path.

We focus on end-to-end losses, because it is the performance parameter for which the benefits of path switching are the most unambiguous [3]. We compute the average loss rate for all paths every 10 seconds (500 samples). If the average loss

TABLE V

THE STATISTICS OF THE 9 POSSIBLE OVERLAY PATHS.

Path #	1	2	3	4	5	6	7	8	9
Avoid (%)	61.9	0.0	37.5	0.0	0.0	0.0	61.9	93.8	100.0
Divergence	2	3	2	3	3	3	2	2	2
RTT (ms)	70	40	10	80	11	87	84	83	270

rate is greater than 1%, we consider the path to be in a loss period. During the entire measurement, 16 loss periods were observed. Although this number is small compared to the entire duration of the experiment, some loss periods exhibit a loss rate as high as 80%. Therefore, path switching can provide significant performance improvements during such periods. For each overlay path, we measure the percentage of avoided loss periods. The results are shown in Table V.

The second row of Table V shows the AS hop at which each overlay path diverges from the default path. The third row shows the round-trip delay from the source node to the relay nodes. If we apply the ED rule, only path 1, 3, 7, 8, 9 remain as potential choices. The round-trip delay on the default path D_{AB} is 30 ms and we apply a delay bound of $\beta = 400$ ms to choose 3 nodes (i.e., $m = 3$) with the largest delay from the source but still below the delay bound (as per Eq. (2)). Based on these criteria only paths 1, 7, and 8 are retained. The actual loss avoidance percentage depends on which one of the three paths is selected, but it has a minimum value of 61.9% and a maximum value of 93.8%, which is significantly better than what a random selection would yield. Note that if a random-4 rule was used, there are several combinations that would produce a poor outcome (0% avoidance in the worst case). It is also notable that although using path 9 can avoid all the loss periods on the direct path, it is not selected by our rule. This is because node 9 is located in Asia and therefore far away from the source and destination nodes. As a result, path 9 might not be able to accommodate the needs of delay-sensitive applications.

VI. CONCLUSIONS

This paper investigates the problem of alternate path selection in a large peer-to-peer environment. It proposes an approach for efficiently selecting a good alternate path from a large number of available candidates. The approach searches for alternate paths that are most disjoint from the default path using AS-level path information. In order to minimize overhead and ensure scalability, a heuristic was introduced (the *earliest-divergence* rule with delay constraints) to select alternate paths using only information that is local to the source node. The effectiveness of the proposed method was demonstrated using a large set of topology data and measurement traces.

An interesting open question is how the performance of the proposed heuristic would change as either the peer-to-peer system or the underlying network grow. In particular, one can distinguish between two different scenarios. One where the network size is fixed, but the penetration of the peer-to-peer system increases, and so does the density of its nodes. The other, where both the network and the peer-to-peer system grow in tandem, so that the density of peer nodes remains mostly constant. One would expect the two to potentially yield ifferent results, as they are likely to differently affect the availability of good alternate relay nodes. Investigating both aspects is the topic of ongoing work.

ACKNOWLEDGMENT

We would also like to thank Professor Zhi-Li Zhang and Sanghwan Lee at the University of Minnesota for their valuable discussion and help. This work is supported by the National Science Foundation under the grants CNS-0085848, ITR-0085824 and ITR-0085930.

REFERENCES

[1] A. Akella, J. Pang, B. Maggs, S. Seshan, and A. Shaikh, "A comparison of overlay routing and multihoming route control," in *Proc. of ACM SIGCOMM*, August 2004.
[2] "RouteScience," http://www.routescience.com/.
[3] S. Tao, K. Xu, Y. Xu, T. Fei, L. Gao, R. Guerin, J. Kurose, D. Towsley, and Z.-L. Zhang, "Exploring the performance benefits of end-to-end path switching," in *Proc. of IEEE ICNP*, October 2004.
[4] S. Tao, K. Xu, A. Estepa, T. Fei, L. Gao, R. Guerin, J. Kurose, D. Towsley, and Z.-L. Zhang, "Improving VoIP quality through path switching," in *Proc. of IEEE INFOCOM*, March 2005.
[5] A. Akella, S. Seshan, and A. Shaikh, "An empirical study of wide-area internet bottlenecks," in *Proc. of IMC*, October 2003.
[6] "Skype," http://support.skype.com/?_a=knowledgebase.
[7] S. Baset and H. Schulzrinne, "An Analysis of the Skype Peer-to-Peer Internel Telephony Protocol," Tech. Rep. CUCS-039-04, Computer Science Department, Columbia University, NY, 2004.
[8] D. G. Andersen, H. Balakrishnan, M. F. Kaashoek, and R. Morris, "Resilient overlay networks," in *Proc. of SOSP*, October 2001.
[9] "Route Views Project," http://www.routeviews.org/.
[10] "RIPE NCC," http://www.ripe.net/.
[11] "PlanetLab," http://www.planetlab.org/.
[12] B. Zhao, L. Huang, J. Stribling, A. Joseph, and J. Kubiatowicz, "Exploiting routing redundancy via structured peer-to-peer overlays," in *Proc. of IEEE ICNP*, September 2003.
[13] A. Akella, B. Maggs, S. Seshan, A. Shaikh, and R. Sitaraman, "A measurement-based analysis of multihoming," in *Proc. of ACM SIGCOMM*, August 2003.
[14] W. Cui, I. Stoica, and R. H. Katz, "Backup path allocation based on a correlated link failure probability model in overlay networks," in *Proc. of IEEE ICNP*, November 2002.
[15] C. Tang and P. K. McKinley, "A distributed multipath computation framework for overlay network applications," Tech. Rep., MSU-CSE-04-18, 2004.
[16] A. Nakao, L. Peterson, and A. Bavier, "A routing underlay for overlay networks," in *Proc. of ACM SIGCOMM*, August 2003.
[17] M. Zhang, J. Lai, A. Krishnamurthy, L. L. Peterson, and R. Y. Wang, "A transport layer approach for improving end-to-end performance and robustness using redundant paths," in *Proceedings of USENIX Annual Technical Conference*, April 2004, p. 99C112.
[18] Z. M. Mao, L. Qiu, J. Wang, and Y. Zhang, "On AS path inference," in *Proc. of ACM SIGMETRICS*, June 2005.
[19] K. Gummadi, H. Madhyastha, S. D. Gribble, H. M. Levy, and D. J. Wetherall, "Improving the reliability of Internet paths with one-hop source routing," in *Proceedings of OSDI*, December 2004.
[20] Z. M. Mao, J. Rexford, J. Wang, and R. H. Katz, "Towards an accurate AS-level traceroute tool," in *Proc. of ACM SIGCOMM*, August 2003.
[21] Z. M. Mao, D. Johnson, J. Rexford, J. Wang, and R. H. Katz, "Scalable and accurate identification of AS-level forwarding paths," in *Proc. of IEEE INFOCOM*, March 2004.
[22] L. Gao, "On inferring autonomous systems relationships in the Internet," *IEEE/ACM Trans. Networking*, vol. 9, no. 6, December 2001.
[23] F. Wang and L. Gao, "Inferring and characterizing Internet routing policies," in *Proc. of IMC*, October 2003.