Exploring the Performance Benefits of End-to-End Path Switching

Shu Tao[1], Kuai Xu[2], Ying Xu[1], Teng Fei[3]
Lixin Gao[3], Roch Guérin[1], Jim Kurose[3], Don Towsley[3], Zhi-Li Zhang[2]
[1]University of Pennsylvania, [2]University of Minnesota, [3]University of Massachusetts

Abstract

This paper explores the feasibility of improving the performance of end-to-end data transfers between different sites through path switching. Our study is focused on both the logic that controls path switching decisions and the configurations required to achieve sufficient path diversity. Specifically, we investigate two common approaches offering path diversity – multi-homing and overlay networks – and investigate their characteristics in the context of a representative wide-area testbed. We explore the end-to-end delay and loss characteristics of different paths and find that substantial improvements can potentially be achived by path switching, especially in lowering end-to-end losses. Based on this assessment, we develop a simple path-switching mechanism capable of realizing those performance improvements. Our experimental study demonstrates that substantial performance improvements are indeed achievable using this approach.

1. Introduction

The widespread deployment of distributed applications is putting renewed emphasis on solutions aimed at ensuring the best possible performance when transferring data between different end-points, but without necessarily incurring the cost and complexity of traditional QoS solutions. In particular, the increasing access to multiple providers and the development of technologies that provide end-users with the ability to control where and how their traffic is to be sent, make it possible to take advantage of "path diversity" to improve the performance and availability of data transfers for such applications. In this paper, we investigate the feasibility and the performance benefits of a mechanism that allows a source end system to dynamically switch among multiple paths to a destination, which is often referred to as "path switching." Our focus and goal is to demonstrate that path switching can indeed deliver meaningful performance improvements in settings involving limited path diversity and by using very simple mechanisms.

The exploitation of path diversity to improve performance has given rise to a number of commercial offerings and research activities that have taken different approaches towards achieving this goal. For example, solutions such as those of [8] and [9] assume that paths between user sites are continuously monitored, and that information gathered via monitoring is used to dynamically select the best provider. Similarly, some providers, e.g., see [7], have opted to offer such a dynamic best path selection to their customers. The potential benefits of those solutions have been partially investigated in [2] with a focus on high volume data sources and data sinks. Alternatively, overlay networks [3] have also been used to exploit path diversity, [4], even in the case that an end system has but a single provider. The investigation of path switching as a mechanism for improving the performance of data transfers has also been motivated by the observations that the default path is often far from optimal, e.g., see [10, 11], and that performance fluctuations can be observed on most Internet paths, e.g., [5, 15].

In general, the ability to improve the performance of data transfers through path switching requires several conditions to be met. First and foremost, there must be sufficient diversity across the different paths over which switching can take place. In other words, performance degradations should not be strongly positively correlated across paths. In our study, we explore this problem via measurements over a wide-area testbed consisting of three separate sites. Path diversity between sites can be achieved either through the use of different providers, or through overlay paths that use one of the sites as a relay point towards the third site[1]. Our investigation reveals that the level of path diversity achievable through either method yields paths with sufficiently decoupled performances, so that path switching has the potential of improving communication performance.

Another requirement for producing meaningful performance improvements through path switching (even when the performance of the various paths is sufficiently uncorrelated), is that the magnitude and time scale of performance variations across paths should both justify and allow track-

[1] See also [4] for an investigation of this issue and [13] for a study of the level of diversity that might be available from a single provider.

ing of the best path. In particular, continuously switching from path to path to track small improvements in performance may be neither feasible nor desirable from an application performance point of view. In order to find out whether these requirements can be met, we conducted extensive measurements on our testbed across a period of several months to estimate both end-to-end delay and loss characteristics of the different paths, thus assessing the potential performance improvement achievable through path switching. Our findings across all those paths consistently showed propagation delay to be the dominant contributor to end-to-end delay, with variations in queueing delays being typically insufficient to change the rank ordering of paths, at least over time scales consistent with path switching decisions, i.e., of the order of a minute. Nevertheless, our findings were somewhat different when it came to end-to-end losses, as the rank ordering of paths was far from stable. Although losses were consistently low across all paths (below 1%), they were not uniformly distributed over time. Because many congestion periods lasted sufficiently long and did not significantly overlap across different paths, path switching had the potential to substantially improve end-to-end loss performance.

Last but not least, switching to a new path is predicated on the assumption that the new path will indeed remain better. This last requirement highlights the need to not only *monitor* the current performance of a path, but to also accurately *predict* its future performance. In other words, a decision to switch to a new path is justified only if the new path outperforms other paths, *after* the switch has occurred. Based on the insight into path behavior derived from our experiments, we developed a simple yet effective methodology for monitoring and predicting path performance and making path switching decisions. The performance improvements offered by this solution were evaluated against those achievable by an "optimal" solution, i.e., a solution that assumes perfect foresight in predicting the best performing path and selecting it. As we shall see, those results show not only that sufficient path diversity exists to achieve substantial performance improvements through path switching, but also that the simple methodology we developed is capable of delivering near optimal performance.

The remainder of this paper is organized as follows. Section 2 introduces the topology of our testbed and the methodology used in collecting measurement data. Sections 3 and 4 are concerned with the characteristics of paths available through multi-homing and overlay solutions, respectively. In both sections, we explore the variations in end-to-end delay and losses observed across different paths, and assess their implications on the benefits of path-switching. Section 5 is devoted to developing an effective path-switching solution for improving end-to-end loss performance. Finally, Section 6 concludes with a summary of our findings.

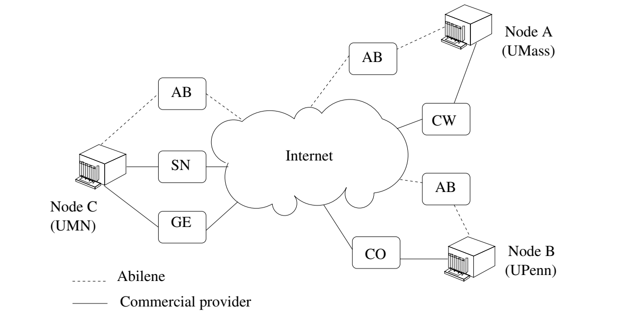

Figure 1. The testbed nodes and their connectivities to different ISP's.

2. Wide-Area Measurement Testbed

2.1. Testbed Setup

In order to explore the benefit of end-to-end path switching, we use a wide-area testbed to apply and validate our analysis. As shown in Fig. 1, the testbed involves three campus networks in the US, two on the east coast (University of Massachusetts and University of Pennsylvania), and one in the Midwest (University of Minnesota), all of which are multi-homed.

To reach other nodes, each node has the ability to select Abilene (AB) or commercial providers, i.e., UMass (node A) via Cable & Wireless (CW), UPenn (node B) via Cogent (CO), and UMN (node C) via Genuity (GE) and Supernet (SN). To enable automatic selection of outgoing ISPs, end hosts in our testbed are assigned multiple IP addresses, and the border gateways are configured with special routing policies: UPenn and UMN use source-address based routing, and end hosts at UPenn/UMN select one of the outgoing ISPs by choosing an appropriate IP address as the source address; UMass installs static routes to other two sites at the border gateway, which selects one of the two outgoing ISPs based on the destination address. In addition, end hosts at each site are also configured with source routing capability to forward traffic. An overlay network is then formed by establishing IP tunnels among them, so that an end host could also use overlay paths to reach other nodes. As a result, there exist more than 10 paths between any given source and destination.

Table 1 gives a snapshot of the AS level paths traversed between end hosts when commercial providers are used. Of the total 13 intermediate ASes covered, four of them are so-called tier-1 ASes (UUNet, Qwest, C&W, Level3) and the others are regional transit networks. The average length of end-to-end AS level path is 4.3 hops. Although our testbed

Src-dst	AS path					
A-B	UMass	C&W	Qwest	Supernet	UMN	
A-C	UMass	C&W	UUNet	UPenn		
B-A	UPenn	Cogent	PSI	C&W	UMass	
B-C	UPenn	Cogent	PSI	Level3	Genuity	UMN
C-A(SN)	UMN	Supernet	Qwest	C&W	UMass	
C-B(SN)	UMN	Supernet	Qwest	UUNet	UPenn	
C-A(GE)	UMN	Genuity	Level3	C&W	UMass	
C-B(GE)	UMN	Genuity	Level3	Yipes	UPenn	

Table 1. The AS level paths between the nodes through commercial providers.

is relatively small, the combination of paths constitute a rich and diverse path set. We believe that this testbed is a representative example for campus or corporate networks, where end-to-end path switching mechanisms may be applied, thus serves as an appropriate setting for our study.

2.2. Measurement Experiments

Using the wide-area testbed, we conducted experiments that continuously monitored and measured the end-to-end performance (in terms of both *delay* and *losses*) of all combinations of paths over a timespan of several months. A measurement daemon is installed at every node on the testbed, where a source node sends time-stamped UDP probes every second to a destination node via different paths. When receiving a probe, the daemon at the destination node records the time at which it is received and stores this along with the source time-stamp in a trace file. From these trace files, we compute the end-to-end delay and loss statistics along various paths. To correlate path characteristics and end-to-end performance, we also run *traceroute* simultaneously (but at a much lower frequency, every 5 or 15 minutes) to track the paths that the probes traverse and record any path change at either the IP (i.e., router) level or the AS level.

Since our objective is to study the benefits of path switching among a set of available paths, we focus primarily on the *relative* performance of those paths, instead of their *absolute* performance. In terms of end-to-end delay, this has the added benefit that clocks at different nodes do not need to be precisely synchronized. To compare the *relative delay* performance among a set of paths between a given source and destination, we select a reference path (e.g., the “best” path over a measurement period) and compute the *difference* between measured delays of other paths and this reference path using data collected in the same probing interval. The loss statistics of each path are computed by counting the number of lost probes over some measurement window. In this paper, we use three sets of measurement traces, denoted respectively as $\mathcal{E}_1$, $\mathcal{E}_2$ and $\mathcal{E}_3$, which were collected from 08/15/2003, 09/02/2003 and 09/15/2003 respectively, each lasting one week.

3. Path Diversity Through Multi-homing

In this section, we analyze and compare the performance of paths via different providers (called *provider paths* in short) on our testbed. This information is helpful in understanding the potential path switching benefits that can be attained *via provider selection*. Since path switching typically incurs a cost (e.g., the application flow could see delay jitter or packet reordering), too frequent path switching is impractical. Therefore, the time scale used in our analysis is no less than 1 minute[2], and we mainly focus on performance variations averaged over minute-long intervals.

3.1. End-to-End Delay Performance

We first focus on the end-to-end delay performance of different provider paths. For a given source and destination node pair and its associated candidate provider paths, we fix a reference provider path and compute the relative delay of other provider paths with respect to this path. Then we rank the relative delay of these provider paths and analyze how the ranking changes over time. Based on our measurement data, our first major observation is that *in terms of end-to-end delay, there usually exists a provider path that almost always outperforms the other provider paths.*

Path	Ranking	Δd (ms)	Path	Ranking	Δd (ms)
A-B(AB)	1	0.0	B-A(AB)	1	0.0
A-B(CW)	2	11.7	B-A(CO)	2	1.5
A-C(AB)	1	0.0	B-C(AB)	1	0.0
A-C(CW)	2	15.8	B-C(CO)	2	22.7
C-A(GE)	1	0.0	C-B(GE)	1	0.0
C-A(AB)	2	7.3	C-B(AB)	2	6.1
C-A(SN)	3	23.2	C-B(SN)	3	28.4

Table 2. Relative delay performance and ranking of provider paths between each source-destination pair.

To illustrate, Table 2 presents the overall ranking of provider paths between every source-destination pair and their (average) relative delays (Δd, with respect to the best performing path) using the dataset $\mathcal{E}_2$. The notation *x-y(z)* indicates that the source is *x,* the destination is *y* and the first-hop ISP from *x* is *z*. Fig. 2 further shows the relative performance gain versus duration for each ranking change when another provider path outperforms the path with the

[2] In Section 5, we further discuss our choice of a 1-minute time scale in our analysis.

overall smallest delay[3]. We see that the majority of ranking changes are short-lived (e.g., less than 1 minute), with mostly small performance gains. Using ranking statistics computed over 1-minute intervals, Table 3 shows (1) the percentage of time that the overall best path indeed provides the smallest delay; (2) the number of ranking changes, i.e., another path outperforms the overall best path; (3) the average duration of ranking changes, i.e., the average duration of another path other than the best overall one having the smallest delay. Between all source-destination pairs, the best overall provider path outperforms other provider paths in more than 99% of time.

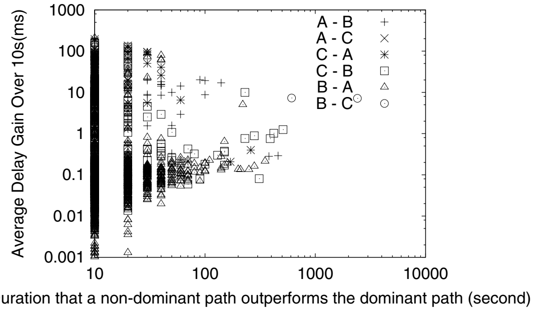

Figure 2. Delay gain achieved by non-dominant paths versus the duration that another path outperforms the best overall path.

Src-dst	Occupation of best overall path(%)	Number of changes	Average duration of rank changes (min)
A-B	99.78	4	3.5
A-C	100.00	0	0.0
B-A	99.64	25	1.4
B-C	99.50	2	25.0
C-A	99.63	6	2.0
C-B	99.65	16	2.1

Table 3. Performance of the overall best provider paths in terms of delay.

To better understand this "best provider path" phenomenon and its generality, we quantitatively consider the various factors that may contribute to the relative delay performance of provider paths and thus their rankings. We first consider the possible effect of queueing delay on provider path ranking changes. For this purpose we partition the measurement traces into 10-minute segments. Assuming that there is no path change during a 10-minute segment, the "queueing delay" is estimated by subtracting the *minimum* delay of the 10-minute segment from the delay measurement data in the same 10-minute segment. If there is a path change in a 10-minute segment, this segment is excluded from the analysis. The results show that queueing delays were relatively small. For example, the probability of queueing delay exceeding 4ms is less than 0.08 on path C-A(AB) [12]. Given that the average relative delay among the provider paths is larger than 7.3 ms (see Table 2), it is evident that queueing delay does not have a significant impact on the relative ranking of these paths.

Since the propagation delay is the dominant factor in determining the relative delay performance of different provider paths, it is natural to ask how path changes will affect their relative performance. To answer this question, we again analyze the traceroute data. We group path changes into two categories: *IP-level* (or router-level) path changes, namely, different IP addresses are seen in the traceroute data for a given provider path, but they still belong to the same AS; and *AS-level* path changes, namely, different AS paths are seen in the traceroute data for a given provider path. The IP address to origin AS mapping is done using BGP information from [1]. From the traceroute data collected over several weeks, we observe that most provider paths are quite stable, which is consistent with observations in [16]. Most IP-level paths last hours before any change occurs and their changes are short-lived, with duration less than 10-15 minutes. In addition, although about 48% of IP-level path changes last no longer than 15 minutes, a large portion of such changes are due to multi-path routing and traffic engineering in certain tier-1 ISPs such as AS 701 (UUNET). Moreover, IP-level path changes rarely affect the relative ranking of the provider paths, as they tend to occur within the same PoP. At the AS-level, the paths show even higher stability. From the data we collected, we found that more than 50% AS paths last at least 12 hours, with a small portion (18.9%) lasting fewer than 15 minutes, which is likely caused by some transient events in inter-domain routing. However, unlike most IP-level path changes, AS-level path changes can have a significant impact on the delay performance of a provider path [12].

We conclude this section by summarizing our major findings. In terms of *end-to-end delay performance*, we find that there exists a best overall provider path that almost always outperforms the others. This is likely because propagation delay is the primary factor that determines the end-to-end delay performance. The difference in the propagation delays of different provider paths comes from the fact that ISP's have different PoP locations and peer with other ASes only at certain PoP's. Queueing delay and IP-level

[3] For clarity, we show only those changes that last more than 10 seconds. The relative delays shown here were the average over 10-second interval, in order to smooth out anomalous delay hikes.

path changes, in general, cause only small delay fluctuations relative to the propagation delay, thus having a minimal effect on the relative delay performance. In contrast, AS-level path changes, albeit rare, may have a significant impact on the relative delay performance. These findings suggest that there is no significant benefit in dynamic path switching for delay performance optimization, in particular, at small time scales (e.g., minutes). Switching to another provider path is only worthwhile when AS-level path changes cause a *significant* and *long-lived* increase in the delay performance of the dominant provider path. Those long-lived delay performance changes, however, can be easily detected without resorting to any sophisticated mechanism.

3.2. End-to-end Loss Performance

We now analyze the potential benefits of path switching for optimizing loss performance by comparing the end-to-end loss rates of different provider paths. From the measurement data collected over our testbed we find that unlike end-to-end delay performance, when it comes to *end-to-end loss* performance there *does not* exist a provider path that consistently outperform others. There are two reasons for this observation. First, as Table 4 shows, the average loss rate of each provider path computed using the dataset $\mathcal{E}_2$ and averaged over the entire duration of the experiments, is extremely low. Second, when losses occur, they tend to come in bursts, and such losses can happen on any provider path. As a result, no provider path consistently outperforms the others.

Path	Loss (%)	Ranking	Path	Loss (%)	Ranking
A-B(CW)	0.0840	1	B-A(AB)	0.1572	1
A-B(AB)	0.1481	2	B-A(CO)	0.5612	2
A-C(CW)	0.0423	1	B-C(AB)	0.2589	1
A-C(AB)	0.0817	2	B-C(CO)	0.8128	2
C-A(GE)	0.3090	1	C-B(SN)	0.0084	1
C-A(AB)	0.3413	2	C-B(AB)	0.0342	2
C-A(SN)	0.7731	3	C-B(GE)	0.0931	3

Table 4. Overall average loss performance and ranking of provider paths.

Although the overall loss rate of each provider path is extremely small, we do observe periods of significant losses on all provider paths that last a few minutes or longer. For example, in Fig. 3 we show the loss rates averaged over 1-minute intervals during a week period starting 09/02/2003 for the two provider paths from node A to node C. We can see many loss "spikes" on both paths with loss rates exceeding 1% or more, indicating that losses on both paths are generally bursty. Moreover, the losses occurring on the two paths do not appear to be highly correlated, as can be inferred from the bottom plot, where the loss rate differences between the two paths are shown. It can also be observed that at times path A-C(AB) has lower loss rates than path A-C(CW), but at other times it is the other way around. Hence no one provider path consistently outperforms the other.

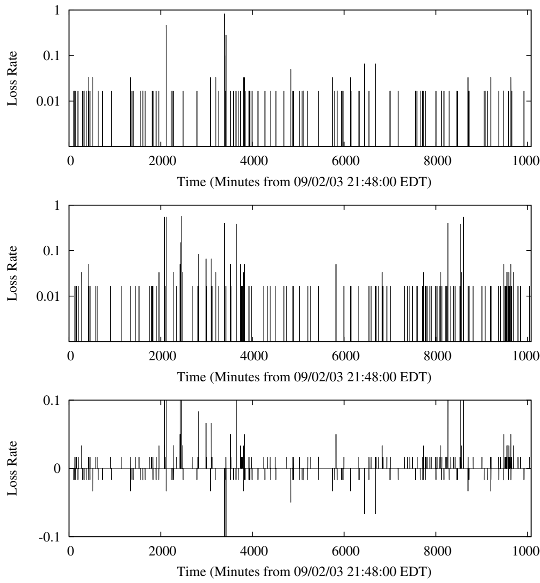

Figure 3. The 1-minute average loss rates on the two provider paths, A-C(AB) (top), and A-C(CW) (middle). The bottom plot shows the relative loss rate difference between the two paths.

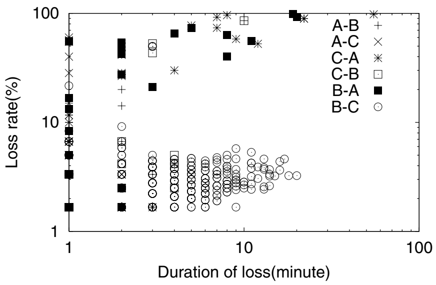

Figure 4. Loss periods of all provider paths.

Fig. 4 illustrates the bursty nature of losses by showing a scatter plot of the loss periods of all provider paths, where the x-axis is the duration of a loss period, i.e., the number of consecutive 1-minute intervals with at least 1 lost probe, and the y-axis is the average loss rate computed over the loss period. From the figure we see that loss periods can some-times last more than 10 minutes, and the average loss rates during such periods can often go above 50% and even reach 100% (some paths actually experienced a few outage periods during which nearly all packets were lost). Intuitively, switching to a different path during those loss bursts can be very beneficial, especially for those points in the upper right quadrant, namely lossy periods of both high loss rate and long duration.

Using the 1-minute average loss rates computed from the dataset $\mathcal{E}_2$, Table 5 summarizes the loss performance comparison of the provider paths for each source-destination pair. The table uses the provider path with the *best overall average loss performance* (the path with rank 1 in Table 4) as the basis for comparison, and shows (1) the percentage of time[4] in which the best path does outperform other provider paths (i.e., occupation of best overall path (%) in Table 5); (2) the number of ranking changes, namely, another provider path outperforms the best path; and (3) the minimum, average, and maximum of the durations over which another provider path has the best average loss performance. Unlike the end-to-end delay performance (see Table 3), the provider path with the best long-term loss performance does not *consistently* outperform other provider paths. As illustrated in the last two rows of Table 5, it is quite possible for the best overall path to only deliver the lowest loss rate for a small fraction of time. This is because having the best overall loss rate over a week-long period does not guarantee that at any given 1-minute interval there are no other paths that offer better performance. This is especially true when multiple alternatives are available, as is the case for site C, which can select among three providers. For example, from node C to node B, the best overall path (i.e., C-B(SN)) outperforms *both* the two other paths (i.e., C-B(AB) and C-B(GE)) for only 2.22% of the time, even though it outperforms each single one of them most of the time. From the above analysis, we can clearly see the potential of path switching that would allow close tracking of the best path in each time interval.

Src-dst	Occupation of best overall path (%)	Number of changes	Duration (minutes) of changes (min, avg, max)
A-B	19.24	318	(1, 1.1, 3)
A-C	60.30	98	(1, 1.1, 2)
B-A	57.46	92	(1, 1.2, 12)
B-C	75.64	657	(1, 1.3, 7)
C-A	4.55	11	(1, 3.8, 22)
C-B	2.22	42	(1, 1.0, 2)

Table 5. Performance of the overall best provider paths in terms of loss.

In general, our findings on end-to-end loss performance of different provider paths suggest that there are potential benefits in performing dynamic path switching at a relatively fine time scale (e.g., a few minutes). To quantify the performance gains we can potentially achieve, we consider an ideal case where the provider path with the best average loss rate over each 1-minute interval is always used, assuming that the average loss rate on each path is known *a priori*. Hence the loss performance using this ideal dynamic path switching reflects the *theoretically best attainable* loss performance of any dynamic path-switching mechanism. Table 6 shows the resulting overall loss rate achieved by the ideal dynamic path switching for all six source-destination pairs (the column marked as "ideal dynamic"). For comparison, the best overall loss performance without path switching (that of the 1st ranked provider path in Table 4) is also shown (the column marked as "best static"). Clearly, the ideal dynamic path switching leads to marked improvement in the overall loss performance for all source-destination pairs.

	Best Static (%)	Ideal Dynamic (%)	Correlation
A-B	0.084	0.013	0.213
A-C	0.042	0.015	0.452
B-A	0.157	0.012	0.020
B-C	0.259	0.079	0.024
C-A	0.309	0.010	0.011/0.026/0.626
C-B	0.008	0.001	0.001/0.016/0.029

Table 6. The achievable loss rate by statically choosing the best path and by ideal path switching.

Lastly, from Table 6 we see that although ideal dynamic path switching attains better overall loss performance for every source-destination pair, the percentage of performance gains is not uniform over all these source-destination pairs. For example, the performance gains of source-destination pairs A-B and A-C can be seen to be much lower than those of the other pairs. This can be explained by considering the loss performance correlation between the provider paths for each source-destination pair. We define the *spatial correlation* between two paths as the correlation coefficient of the 1-minute average loss rates between them. By computing this spatial correlation coefficient between two provider paths of each source-destination pair, we find that except for three pairs of provider paths, the correlation coefficient for all other pairs of provider paths is less than 0.03 (see Table 6). For the two provider paths from node A to node B, the correlation coefficient is about 0.21, and for the two provider paths from node A to node C, it is about 0.45. This mild loss correlation limits somewhat the potential performance gains of ideal dynamic path switch-

[4] In most 1-minute intervals, the loss rates are 0 on all the paths. Hence, we only count those intervals in which the loss rate of the best overall path is not equal to that of the other path(s).

ing. In the case of node C to node A, the two provider paths via SN and GE have a correlation coefficient of about 0.63. However, losses on these two provider paths are not correlated with those of the third provider path via AB. Consequently, the ideal dynamic path switching is still able to achieve over 95% performance improvement by switching to the third provider path when the other two paths experience losses.

In summary, we have shown that as long as losses on the candidate provider paths are not strongly correlated, dynamic path switching based on selecting the best performing provider path over, say, a one minute time scale, can potentially offer meaningful gains in end-to-end losses.

4. Path Diversity Through Overlays

Overlay networks can also provide path diversity. From the end-user's perspective, there is no difference between an *overlay* path and a *direct* (provider) path. However, unlike direct paths going through different ISP networks, overlay paths tend to share more common segments with the underlying direct paths used to form the overlay. This can lead to stronger correlation between overlay and direct paths, and as a result marginal potential for performance improvement from path switching. In the previous section we also observed that path diversity was of limited benefit when it came to improving end-to-end delays. This remains true with overlays, hence, we restrict our investigation to examining the potential that overlays have for improving end-to-end loss performance.

In order to eliminate the influence of multi-homing, we select one provider for each site and consider the resulting overlay network spanning the three sites. Using our testbed, there are twelve possible overlay networks via different combinations of provider selections. This produces an overlay network with only two candidate paths for each source and destination pair: the direct path via the selected provider and the "two-hop" overlay path via the third node.

	Dir. (%)	Dyn. (%)	Corr.	Dir. (%)	Dyn. (%)	Corr.
A-B	0.084	0.033	0.482	0.148	0.138	0.993
A-C	0.042	0.026	0.867	0.082	0.057	0.400
B-A	0.157	0.017	0.032	0.157	0.138	0.547
B-C	0.259	0.006	0.007	0.259	0.008	0.002
C-A	0.309	0.009	0.042	0.341	0.138	0.401
C-B	0.034	0.000	0.001	0.093	0.088	0.513

Table 7. Loss rate improvement achieved by ideal dynamic path switching between direct and overlay paths in the first case study (left) and in the second case study (right).

We initiate our study of end-to-end loss performance in an overlay with two case studies. In the first case, nodes A, B, and C use CW, AB, and GE as their respective provider. In the second case, they all use AB as their provider. We compare the loss performance of the direct path to that obtained by dynamically selecting every minute the better path, direct or overlay, for each source-destination pair. As shown in Table 7, the loss rate improvement from the additional overlay path is remarkable in the first case study. However, the improvement is less obvious for the second case study (also see Table 7). By computing the spatial correlation of each path pair in the above two cases, we found that when the three sites are all connected through AB, the candidate paths are more likely to be strongly correlated. This might be explained by the fact that the overlay network using Abilene as the provider shares more physical links. From Table 7, we also observe that there exist certain source-destination pairs for which the potential performance improvements are rather limited. For example, in both cases the path pairs originating from node A have strong correlations, which greatly limit the resulting performance improvement.

So far we have observed that a significant loss performance improvement is achievable when spatial correlation between the direct and overlay paths is small. We investigate next whether the spatial correlation factor can serve as a *qualitative* measure for predicting potential performance improvement. Consider, a pair of paths P_1, P_2 with overall loss rate s_1 and s_2, and let s' denote the loss rate obtained by dynamically selecting the path with better loss performance in an off-line fashion. Let $R(P_1, P_2) = (\min(s_1, s_2) - s')/\min(s_1, s_2)$ denote the best attainable loss performance improvement relative to the best single path. In Fig. 5, the relationship between the spatial correlation and the achievable loss performance improvement is shown for all possible pairs of direct paths and all direct-overlay path pairs. We observe from Fig. 5 that the connection between spatial correlation and performance improvement is rather strong. For example, if a pair of paths has spatial correlation lower than 0.2, the best attainable performance gain from path switching is always larger than 70%. Conversely, if a pair of paths exhibit strong correlation in their loss processes, then the likelihood that they benefit significantly from dynamic path switching is also greatly reduced. However, there are still cases where the potential performance improvement is larger than 40%, even for path pairs with a spatial correlation factor larger than 0.6.

To summarize, we have observed that path switching also has the potential for improving end-to-end loss performance even within an overlay. In some cases, this improvement is not as large as what is achievable through multi-homing based path switching. This is most likely due to the presence of overlaps between direct paths and overlay paths. In the next section, we explore practical predictor-

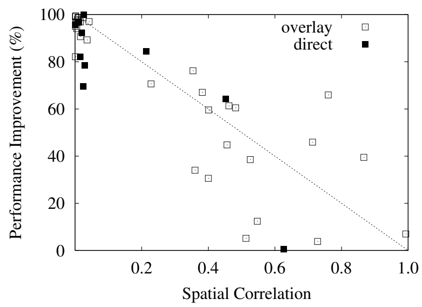

Figure 5. Performance improvement and spatial correlation.

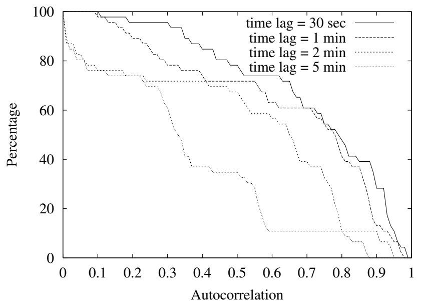

Figure 6. The percentage of traces whose autocorrelations exceed different values.

based techniques that can achieve much of the potential improvement, whether through multi-homing or through overlays.

5. The Case for Path Switching

Our previous measurement study demonstrated that there is significant potential for reducing end-to-end loss rates through path switching. In this section, we show that this potential can be realized in practice. In particular, our previous assessment relied on an ideal model where we always knew ahead of time which path was going to be the best one. This is obviously not a realistic assumption, and the first step towards realizing the potential benefits of path switching is, therefore, to develop a practical and effective method for predicting path performance and more specifically which path will offer the best performance. In the next two sub-sections, we focus on exploring if and how this is feasible.

5.1. Predictability of Path State

A random process is predictable only if it exhibits some form of temporal dependency. Our analysis of the traces that we have gathered shows the presence of temporal correlation. We computed the autocorrelation functions with different time lags for all 46 traces in $\mathcal{E}_2$, based on their average loss rates. The average loss rate is computed every minute when the time lag is greater than or equal to 1 minute, and is computed every 30 seconds when the time lag is 30 seconds. Fig. 6 shows the percentage of traces whose autocorrelation functions exceed different values. It can be observed that when the time lag is 30 seconds or 1 minute, most of the traces show strong temporal correlation, and the correlation decreases as the time lag becomes larger. However, even with a time lag as large as 5 minutes, over 35% of the traces still have autocorrelations exceeding 0.5. This suggests the possibility of using past path loss behavior to predict future path loss behavior.

Markov processes are capable of capturing correlation and can, therefore, be used to predict future loss behavior (for example, see [6] for an application to a related problem of predicting degradations in round-trip delays, and using that information to select an "exit" gateway). In predicting loss, a Markov model can be tuned to operate at any time scale. From the data of Fig. 6, we know that in general smaller time scale offers higher temporal correlation when analyzing loss. However, the temporal correlation in loss does not significantly increase when the time lag is reduced from 1 minute to 30 seconds. Since path switching and the associated re-routing incur a cost, using too fine time granularity for performance prediction and path switching has disadvantages. We therefore select 1 minute as the time scale of the Markov model used for path state prediction.

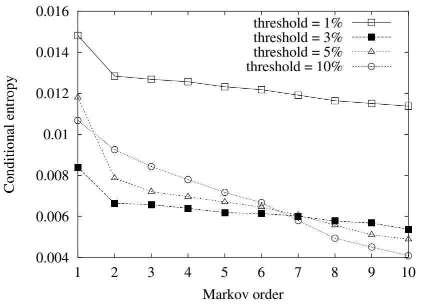

Figure 7. The effect of the order of Markov model k and the threshold θ on the predictability of path state.

We begin by labeling a path as being in one of two states, either good or bad. Suppose we use an observation interval of length w, the state of path X is defined as 1 if the average loss rate in the interval is greater than θ, and 0 otherwise. We use a kth order Markov process to represent the temporal correlation of the path state; the process is defined by the probability

$$\begin{aligned} Pr\{X_i = x_i | X_{i-1} = x_{i-1}, ..., X_{i-k} = x_{i-k}\} \\ = P(x_i | x_{i-1}, ..., x_{i-k}). \end{aligned} \quad (1)$$

where X_i (x_i) refers to the state of the path in interval i.

The order of the Markov model, k, determines how many intervals need to be observed before we can predict the state of the path in the next interval. It then remains to choose the parameters w and θ. As mentioned earlier, we choose w to be one minute because of the strong correlation that we observed in the traces at that lag. In order to choose appropriate values for θ and k, we focus on the *predictability* of process X as a function of these parameters. Let $\overline{b}$ represent the observed sequence $(x_{i-1}, x_{i-2}, ..., x_{i-k})$, and a represent the value of X_i, then the predictability can be measured by the empirical conditional entropy (see [14] for details)

$$H(\theta, k) = -\sum_{\overline{b} \in X^k} \Pr(\overline{b}) \sum_{a \in X} P(a|\overline{b}) \log P(a|\overline{b}). \quad (2)$$

where $X^k = \{0, 1\}^k$ is the sample space of $\overline{b}$. The empirical conditional entropy is 0 if X is completely predictable and $\log |X|$ if X is completely random (note that $P(a|\overline{b})$ could be 0, hence we define $\log 0 = 0$). Given a time series representing process X, we can count the number of times that state $\overline{b}$ is observed ($l^k(\overline{b})$), as well as the number of times that state $\overline{b}$ is followed by state a ($l^k(a, \overline{b})$). Then $P(a|\overline{b})$ can be estimated as

$$P(a|\overline{b}) = \begin{cases} \frac{l^k(a,\overline{b})}{l^k(\overline{b})}, & l^k(\overline{b}) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

We estimated this value for the C-A(AB) path using the one-week $\mathcal{E}_1$ trace collected on 08/15, and with different values of k and θ. From the results shown in Fig. 7, we observe that:

- Predictability is slightly improved by increasing the order of the Markov predictor. This suggests that there are some long term temporal correlations in this loss process; thus having a longer memory of the history may lead to a more accurate prediction.
- The selection of θ has a critical effect on the predictability of loss. An appropriate value of θ improves the predictability of path quality because it enables better differentiation of the path states. For instance, $\theta = 3\%$ seems to be the best choice for this particular trace.

Although some of the above conclusions are specific to this example, we observed similar trends when studying the other traces. Henceforth, we use $\theta = 3\%$ as the threshold to define the 2 states of our path model. We further study the effect of the order of the Markov model on prediction accuracy in the next section.

5.2. Quality of Prediction

The performance of a predictor can be measured by the *precision* rate p, the fraction of predicted states that match the observed states, and the *recall* rate r, the fraction of observed states that are correctly predicted. Let h_t represent the prediction of X in interval t, x_t denote the actual value of X in interval t, then p and r can be defined as

$$p = P(X_t = x_t | h_t = x_t) = \frac{P(h_t = x_t, X_t = x_t)}{P(h_t = x_t)}. \quad (4)$$

$$r = P(h_t = x_t | X_t = x_t) = \frac{P(h_t = x_t, X_t = x_t)}{P(X_t = x_t)}. \quad (5)$$

Clearly, a good predictor should have both high precision and recall rates. Because the good state is dominant throughout all traces, the prediction accuracy for the good state ($> 95\%$) is much higher than for the bad state. Therefore, we focus on the precision and recall rates of different predictors in predicting the bad state.

We first study a simple predictor, which always predicts the state in the next interval as the state in the currently observed interval, i.e.,

$$X_i = x_{i-1} \quad (6)$$

This predictor assumes that the state of a path does not change in one observation interval. Note that the precision rate and the recall rate of this simple predictor are the same. We also study Markov predictors of different orders, i.e., given that the states of a path in the last k intervals, $X_{i-1}, X_{i-2}, ...X_{i-k}$, are $\overline{b}$, we predict its state in the next interval i as

$$X_i = \arg\max_a P(a|\overline{b}). \quad (7)$$

The advantage of a Markov predictor over the simple predictor is that it relies on history to make a prediction. The first-order model maintains little history, only one observation interval, and thus does not exhibit this advantage. In fact, in most cases, the first-order Markov predictor makes the same decision as the simple predictor.

Higher-order models can account for long-term correlations in loss patterns. However, this requires that the temporal dependencies of the observation state sequence be stationary, which may not be the case in a real trace. As a result, while having longer memory can potentially increase the recall rate of the Markov predictor, it may also lower the precision rate. We measured the overall precision and recall rates for all traces in dataset $\mathcal{E}_2$, using both the simple predictor and Markov predictors of different orders. The results are shown in Table 8. For this dataset, the 4th-order Markov model gives the best overall performance. However, the improvements of both precision and recall are very small compared to the simple predictor. Note that the average precision/recall rate is not very high. This is because many traces contain only a small number of bad states, and these bad states are typically not temporally correlated. To predict such sporadic loss events *ahead of time* is nearly impossible. It therefore makes sense to consider the performance of a predictor once a path has entered a bad state, i.e., a conditional performance measure. The last column of Table 8 shows the recall rate of the different predictors conditioned on the fact that the predictor has *at least* one bad state in its memory. As can be seen from the table, this conditional accuracy of the predictors is much higher. Note that the higher-order models have a lower conditional recall rate. This is because their longer memory, combined with the condition of having at least one bad state in memory, introduces larger possibility of incorrect predictions. In contrast, both the simple predictor and the 1st order predictor achieve a 100% recall rate as they essentially always predict another bad state after experiencing the first bad state.

The 2-state path model provides the simplest classification of path quality, but its coarseness can hide differences in the quality of two paths. For example, if one path has a loss rate $s_1 = 0$ and another has a loss rate $s_2 = 2.5\%$, they are both considered to be in a "good" state when using a 2-state path model with $\theta = 3\%$. Categorizing loss rates into a larger number of states provides a finer granularity definition of path state. However, this need not improve prediction accuracy, because the addition of new states increases the probability that the model makes an incorrect prediction. For example, we can classify path quality using a finer granularity as being "good", "acceptable", "bad", or "very bad", based on three thresholds, say, $1\%, 3\%$, and 5%. The last row of Table 8 shows the results of applying a first-order Markov predictor to this 4-state path model for all the traces. Note again that the precision-recall results are only computed for the lossy states, i.e., $1\% \leq s < 3\%$, $3\% \leq s < 5\%$ and $s \geq 5\%$, and that the conditional recall rate is computed given that one of the 3 lossy states is observed in the last interval. The overall precision and recall percentages clearly show that the probability of correctly predicting the lossy states decreases as the number of states increases. However, in spite of its lower absolute accuracy, a finer definition of path state could still be beneficial when comparing the *relative quality difference* between two paths, which might help make better path switching decisions. As we will see in the next section, the results of this trade-off are case-dependent.

Predictor	Precision (%)	Recall (%)	Cond. Recall (%)
Simple predictor	34.82	34.82	100.0
1st-order Markov	34.81	34.86	100.0
2nd-order Markov	34.80	34.88	74.41
3rd-order Markov	34.87	35.25	64.46
4th-order Markov	34.87	35.37	58.82
5th-order Markov	34.85	35.46	55.11
6th-order Markov	34.71	35.78	52.88
4-state 1st-order	26.33	26.33	62.11

Table 8. The performance of prediction with different models.

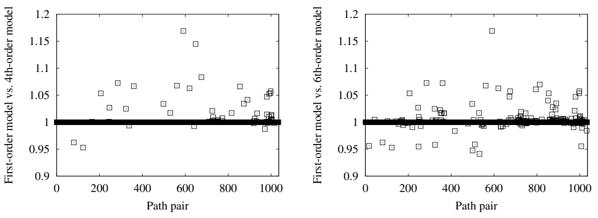

Figure 8. Comparing the resulting loss rates of using first-order, 4th-order and 6th-order Markov predictors.

5.3. Prediction-Based Path Switching

Based on the above analysis, we design the following path switching strategy. First, we use a predictor to predict the state of each path in the next time interval (1 minute). Then, at the beginning of each time interval, we choose the candidate path with the best predicted state. We remain on the current path, unless a better path exists.

We first compare the loss rates produced by the the above strategy when using Markov predictors of different orders. Fig. 8 compiles the ratios of the loss rate of the first-order Markov predictor to those of the 4th-order and 6th-order Markov predictors, for all combinations of traces and source-destination pairs. The figure confirms the slightly better performance of the 4th-order predictor, but shows that although the 6th-order predictor decreases loss rate in some cases, it also yields worse performance in some other cases. This is because although the 6th-order predictor has a higher recall percentage, it has a lower precision percentage than the first-order predictor. For both the 4th-order and the 6th-order predictors, the loss rates are improved only for a few trace pairs. This suggests that a higher-order predictor in many cases does not provide a significant advantage over a simple (first-order) predictor. Therefore, we focus on the latter in the rest of this section.

To evaluate the performance of our path switching mechanism, we compare its performance to that of the ideal switching scheme that has perfect knowledge of future path

Trace-Pair	1	2
Loss rate on path 1 (%)	0.24	0.33
Loss rate on path 2 (%)	0.12	0.30
Best attainable loss rate (%)	0.003	0.13
Loss rate for 2-state model (%)	0.027	0.17
Error by mis-prediction (%)	0.024	0.04
Loss rate for 4-state model (%)	0.024	0.19
Error by mis-prediction (%)	0.021	0.06

Table 9. The performance of prediction-based path switching using different models.

states and always picks the best. We use two trace-pairs, both of length one week, as examples. As shown in Table 9, the best attainable loss rate is 0.003% for trace-pair 1, composed of two traces with average loss rates 0.24% and 0.12%. This value is 0.13% for trace-pair 2, consisting of two traces with loss rate 0.33% and 0.30%. We first assume a 2-state path model, for which results are given in the fourth row of Table 9. Although there is a gap between the resulting loss rate and that of ideal path switching, the prediction-based mechanism still reduces loss rate significantly. We then compare this result with what is achieved when using a 4-state path model. For trace-pair 1, the 4-state model performs slightly better than the 2-state model. This implies that, although the 4-state model is less accurate in predicting path state, its ability to differentiate path quality at a finer granularity helps path switching decisions. However, for trace-pair 2, this advantage is overtaken by the fact that the 4-state model tends to result in more incorrect predictions. We used both the 2-state and the 4-state path models for all trace-pairs in our datasets, and the resulting loss rate ratios are shown in Fig. 9. It can be observed that the ratios are mostly close to 1, so that neither path model exhibits a clear advantage.

For both the 2-state and the 4-state models, the loss rate is higher than that of the off-line reference, and it is important to understand the reasons behind this difference. The first-order predictor we are using has an intrinsic limitation in its ability to predict the state of a path, and therefore allow timely switching decisions. It needs at least one observation interval to detect the onset of congestion, so that path switching decisions are always off by one interval (1 minute). Conversely, path switching experiences a similar one interval lag when a path's state goes from bad to good so that it now becomes a better option, i.e., the other path is also in a bad state. A key question is, therefore, whether or not the losses that occur during those transition intervals account for the difference in performance with the ideal off-line model. We call these essentially "unavoidable" errors, mis-prediction errors, and investigate their magnitude by accounting for all the losses that take place during such periods. The results shown in Table 9 indicate that this is indeed the main cause for the difference with the ideal off-line model. Bridging that gap appears, therefore, impossible.

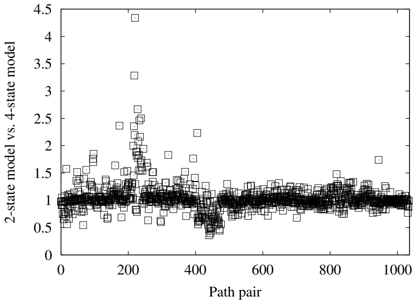

Figure 9. Comparing the resulting loss rates using 2-state and 4-state path models.

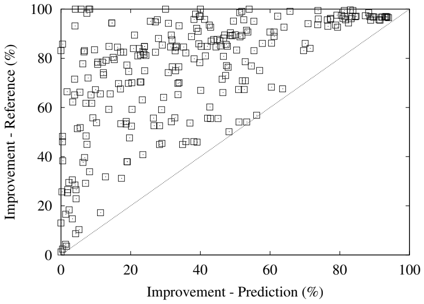

Figure 10. The relative loss rate improvement achieved by prediction-based path switching, compared with the off-line reference.

In Fig. 10, we extend our comparison from two trace-pairs to all possible pairs of paths in our testbed. This includes direct multi-homing paths and overlay paths. As before, the performance of the path switching decision is compared to that of an ideal off-line decision. Each point in the figure corresponds to a different pair of paths. The loss rate improvement is a relative value. Namely, for a pair of candidate paths with loss rates s_1 and s_2, we first compute the resulting loss rate of prediction-based path switching s', and the relative improvement is then computed as $\frac{\min(s_1,s_2)-s'}{\min(s_1,s_2)}$. The reference (optimal) value is computed similarly. Fig. 10 shows that in a few cases the loss improvement achieved by the on-line path switching mechanism is fairly close to the best attainable value, while in many other cases it is not. However, for most trace pairs the improvement in loss performance remains substantial. The issue of whether multi-homing or overlay paths provides a greater opportunity for improvement is explored in Fig. 11, in which each point corresponds to a pair of *direct-direct* or *direct-overlay* paths. The comparison results show that in most cases in our environment there does not appear to be a major difference between the two.

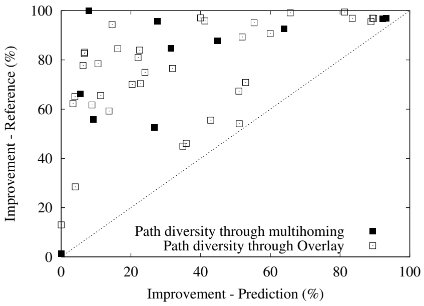

Figure 11. The relative loss rate improvement achieved by prediction-based path switching, with path diversity through multi-homing or overlay, compared with the reference.

6. Conclusions

The increasing availability of path diversity when connecting multiple end-points makes it possible to consider improving communication performance simply by taking advantage of the fact that not all paths experience poor performance at the same time. In this paper, we investigate the feasibility of this idea in a reasonably representative setting, and devise a simple path switching mechanism to demonstrate the performance improvements that can be achieved. Our study shows that it is possible to achieve reasonable path diversity by relying either on limited multi-homing or through overlay-routing. In particular, we find that when it comes to losses, a relatively small amount of path diversity appears capable of producing paths with non-overlapping loss periods and, therefore, offers the opportunity to improve performance through path switching.

7. Acknowledgments

This work is supported by the National Science Foundation under the grants ANI-9906855, ANI-0073819, ITR-0085930, and ITR-0085824.

We also would like to thank Peter Bartz, Frank Di Gravina, David Farmer, and Chris Hertel at UMN, Tyler Trafford at UMass, and the ISC department at UPenn for their support and assistance in setting up the testbed.

References

[1] Route views project. http://www.routeviews.org/.

[2] A. Akella, B. Maggs, S. Seshan, A. Shaikh, and R. Sitaraman. A measurement-based analysis of multihoming. In *Proc. of ACM SIGCOMM*, August 2003.

[3] D. G. Andersen, H. Balakrishnan, M. F. Kaashoek, and R. Morris. Resilient overlay networks. In *Proc. of SOSP*, October 2001.

[4] D. G. Andersen, A. C. Snoeren, and H. Balakrishnan. Best-path vs. multi-path overlay routing. In *Proc. of Internet Measurement Conference*, October 2003.

[5] J. Bolot. End-to-end packet delay and loss behavior in the Internet. In *Proc. of ACM SIGCOMM*, September 1993.

[6] A. Bremler-Barr, E. Cohen, H. Kaplan, and Y. Mansour. Predicting and bypassing end-to-end Internet service degradations. *IEEE J. Select. Areas. Commun.*, 21(6):961–978, August 2003.

[7] Internap. http://www.internap.com/.

[8] ProficientNetworks. http://www.proficient.net/.

[9] RouteScience. http://www.routescience.com/.

[10] S. Savage, A. Collins, E. Hoffman, J. Snell, and T. Anderson. The end-to-end effects of Internet path selection. In *Proc. of ACM SIGCOMM*, September 1999.

[11] N. Spring, R. Mahajan, and T. Anderson. Quantifying the causes of path inflation. In *Proc. of ACM SIGCOMM*, August 2003.

[12] 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. Technical report, http://www.cs.umn.edu/research/networking/itr-qos/papers/, 2003.

[13] R. Teixeira, K. Marzullo, S. Savage, and G. M. Voelker. In search of path diversity in ISP networks. In *Proc. of Internet Measurement Conference*, October 2003.

[14] M. Yanjnik, J. Kurose, and D. Towsley. Packet loss correlation in the MBone multicast network: experimental measurements and markov chain models. Technical report, UMass CMPSCI TR#95-115.

[15] M. Yanjnik, S. Moon, J. Kurose, and D. Towsley. Measurement and modeling of the temporal dependence in packet loss. In *Proc. of IEEE INFOCOM*, March 1999.

[16] Y. Zhang, V. Paxson, and S. Shenker. The stationarity of Internet path properties: Routing, loss, and throughput. In *ACIRI Technical Report*, May 2000.