On Internet Availability: Where Does Path Choice Matter?

Vijay Vasudevan, David G. Andersen, Hui Zhang
Carnegie Mellon University

March 30, 2009
CMU-CS-09-114

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

Today's Internet availability is low, despite the efforts of organizations to improve failure resilience through multi-homing. In this paper, we analyze where and how much exposure to topology is needed to best improve availability without sacrificing scalability. Through this analysis, we find that exposing choice of the first and last AS hops between multi-homed AS pairs can almost always provide a maximum number of AS disjoint paths through the network, whereas the paths exposed under multi-homing tend to share the same inbound route to multi-homed destinations.

We qualify our topological analysis with an active measurement study showing that exposing choice of the first and last AS hops can provide availability approaching the optimal availability possible, suggesting that high availability can be obtained without sacrificing scalability.

Keywords: Internet architecture, network, interdomain routing, availability, AS topology, planetlab, measurement

1 Introduction

Today's Internet availability is low compared to the telephone and emergency service networks [25]. Techniques such as multi-homing can improve availability, but not always to the degree that a site might desire [22, 1].

Fundamentally, network availability is determined by three properties: 1) *responsiveness*: the granularity of failure detection and the speed of recovery; 2) *topological richness*: the physical redundancy present in the Internet; and 3) *topological exposure*: the portion of the topology that is exposed for use in selecting paths.

Responsiveness: Numerous studies have described ways to make routing protocols respond more rapidly to failures [24, 21, 8]. Others have shown that allowing end-system input in path selection can improve the granularity of failure detection beyond simple link and node failures [4, 3, 19, 38].

Topological Exposure: Yet more research has shown that exposing more of the topology for path selection, e.g. in the form of multi-homing or overlays can also improve availability [1]. Several studies provide *mechanisms* for obtaining and using multiple paths [44, 46, 3, 19, 38, 1, 4, 24, 21], each using differing degrees of topological exposure. The effectiveness of any path selection mechanism depends critically on which portions of the topology it can choose from. This work is agnostic to the exact mechanism used, assuming only that one exists. In this paper, we attempt to understand how much availability can be achieved by exposing different sets of Autonomous System (AS) paths.

On one end of the exposure spectrum lies single path routing (using the Border Gateway Protocol – BGP), where only one path to a prefix is exposed. On the other end lies source routing, which exposes all paths but suffers from scalability problems. The essential tradeoff in this spectrum balances the level of exposure to topological information with the scalability problems introduced from having to prune paths.

Through Autonomous System (AS) topology measurements, we find that exposing the topology of the edges of the network provides the most effective improvements to availability: most multi-homed pairs can obtain a maximum number of disjoint paths through the Internet core by *only choosing the first and last AS hop on the path*, allowing BGP to route from multi-homed provider to multi-homed provider.

In contrast, we discover that multi-homing usually fails to provide the necessary exposure required to produce disjoint paths through the network, due in part to unequal AS path lengths and the tradeoff of low route exposure for improved scalability under BGP.

We qualify this analysis with an active measurement study, measuring availability to real multi-homed destinations on the Internet. We find that exposing the first and last hops between multi-homed ASes can significantly improve availability over simple multi-homing, completely eliminating failures for 10-20% of paths that could not be avoided by multi-homing alone. In addition, choosing the first and last hops provides availability that approaches the optimal availability possible.

We believe these results provide important guidelines for current and future multi-path architectures designed to improve availability, by suggesting where topological exposure is most important and by demonstrating using real-world measurements that path disjointness can improve availability. Moreover, these results indicate that the topological redundancy introduced by multi-homing can provide resilience to failure *if* end-hosts are exposed to the right set of paths.

The measurement methods described in this paper present several techniques for discovering and verifying multi-homed end-hosts on the Internet. Our data collection and analysis techniques give a method of evaluating the availability of paths that traverse routers off the default path, allowing measurement of paths beyond those currently provided by BGP.

Our paper is organized as follows: §2 provides a background on availability, BGP routing and its

drawbacks with respect to multi-homing. §3 describes our study of path disjointness on inferred AS topologies. §4 and §5 describe our active measurement methods and results, respectively. Lastly, we conclude with a brief discussion of architectural implications and related work in §6.

2 Background

End-to-end availability on the Internet is low compared to the telephone and emergency service networks [4, 13]. Unfortunately, many different factors contribute to reduced availability, not one easily-fixed flaw:

1. **ROUTING CONVERGENCE** [25]: BGP may take from several to tens of minutes to converge to new paths after a failure. Stability controls such as route flap damping can make the problem worse [31].

2. **MISCONFIGURATION** is a major source of routing problems [30], at the router, BGP, or IGP level. Many operators still manually configure each router [9], making human error a large factor. Even if current efforts to verify [14] or centralize [9, 18] router configuration become widely deployed, human error is likely to remain a major cause of outages.

3. **MALICE AND SECURITY VULNERABILITIES**: DoS attacks, prefix hijacking, and other attacks against the infrastructure pose significant threats to availability [7, 32]. While proposals exist to secure routing or thwart DoS attacks, such proposals have not yet been adopted widely enough to realize their benefits.

4. **HARDWARE AND SOFTWARE BUGS** and other failures that affect end-to-end connectivity without being reflected in the control-plane are another significant source of downtime [37]. Unfortunately, when such flaws materialize only at the data-plane, routing cannot circumvent them, making such problems harder to diagnose and remedy.

2.1 Host-informed Path Selection

Many recent research efforts have focused on improving end-to-end Internet availability and performance [38, 1, 4, 19]. Their results overwhelmingly suggest that some form of host-informed path selection can enhance availability and performance beyond that provided by current intra-domain and inter-domain protocols. Several proposals for failover-based inter-domain routing show how to quickly recover from failures, but require that failures be detected by the routing plane [8, 24, 21]. In contrast, host-informed path selection provides failure detection at a finer, application-specific granularity.

For example, both Akella et al. [1] and MONET [4] enable end-hosts to select from multiple paths for better availability and performance. Their results suggest two principal reasons for allowing end-hosts to participate (to some degree) in path selection.

First, end-to-end path selection masks failures across a wide array of components, regardless of the cause or precise location of those failures. With today's Internet outages having varied and complex causes (e.g., misconfiguration), host-informed path selection is an appealing way to mask those failures without confronting the myriad ways in which individual components can fail.

Second, path availability is ultimately application-specific. Instant messaging applications may be satisfied with low bandwidth channels and 3 second latencies, but interactive voice or video applications

would certainly feel otherwise. Regardless of its specific needs, the application is the ultimate arbiter of whether or not communication is successful.

While providing end-hosts with multiple paths can be effective, providing too many path choices can be paralyzing, particularly when many paths must be probed before succeeding [46]. We believe a multi-path architecture should present a small set of paths that are likely to be high-performing and failure disjoint. Regardless of *how* paths are selected, in this paper we attempt to identify *where* path choice should be provided to end-hosts to support this goal.

2.2 On Multi-homing

> "Most surprisingly, we discovered many cases in which an origin AS was unreachable through one of its providers but not others, suggesting that multi-homing does not always provide the resilience to failure that it should." – E. Katz-Bassett [22]

The most popular form of obtaining multiple paths today is through multi-homing, whereby an organization obtains connectivity to the Internet through several providers. Previous research has shown that for multi-homed sources contacting single-homed destinations, availability is limited by failures near the single-homed destination [1]. Unfortunately, the recent quotation above suggests that *multi-homed destinations* do not observe the failure resilience expected by multi-homing.

To understand this lack of resilience, we begin by briefly describing how BGP advertises a multi-homed organization's inbound routes. Next, we explain the how the limitations of inbound route selection can lead to decreased failure resilience.

Inbound route advertisement in BGP: A multi-homed organization's providers will announce that they can reach the organization's IP prefix(es) to neighboring ASes using BGP. When another AS hears of two or more paths to the same prefix, it will forward *only one* of these paths to neighboring ASes. While the choice of which path to forward is up to the forwarding AS, most ASes tend to forward the shortest AS path. As a result of BGP's forwarding policies, nearly all ASes only hear of a subset of the paths to a given prefix.

Limitations of inbound route selection: For outgoing packets, multi-homing can provide resilience to nearby failures using some form of host-informed path selection (e.g. intelligent multi-homing [1]): if one of the source's providers fails, the source can route outgoing packets through one of its working providers. Incoming packets, unfortunately, are not as easily controlled by the source: a destination replying to the multi-homed source relies on BGP to provide it with a working path.

If all AS paths advertised to the destination route through a failed provider, the destination will not be able to directly send packets to the source. Until the path repairs or the source's working path announcement reaches the destination, the two will be unable to bilaterally communicate. Unfortunately, the delay between a detected failure and the destination hearing the announcement of the new working path can often take minutes to hours, significantly reducing availability [25]. Prior work has overcome this limitation using NATs or by proxying all traffic through intermediate nodes [4, 19].

Exposing choice at the edges: The results of work on end-host path selection systems suggest that organizations benefit from first hop path choice because most failures occur at the network edges [1, 4].

Explicitly providing path choice at both the source and the destination edges can increase control of inbound route selection and introduce path diversity where failures are most likely to occur. As we show later in the paper, providing first *and* last hop choice to hosts appears quite effective in circumventing most failures (§5), while also providing a principled explanation for why these paths should be failure disjoint (§3.2).

3 Analysis of Path Disjointness in the AS Topology

To understand where exposure to topology can most benefit availability, we perform an analysis of failure resilience on inferred AS topologies. As a first approximation of failure resilience, we calculate the number of AS disjoint paths between multi-homed AS pairs using measurements on several inferred AS topologies.

The result of this analysis is that most paths entering a multi-homed destination tend to route through a subset of the available providers of that destination, yielding suboptimal AS path disjointness for most AS pairs. By explicitly exposing path choice of both the first and last AS hops, most multi-homed AS pairs can obtain a maximum number of disjoint paths while BGP determines routes through the core of the Internet.

3.1 Metric of Choice: AS Path Disjointness

We choose AS path disjointness as a representative metric for evaluating the failure resiliency of multi-homing in this section. Path disjointness is a useful metric for evaluating failure resiliency for the following reasons:

Access links frequently limit availability [1]. Hardware used at the edges is often less reliable or less expertly managed. The cost of providing physical redundancy is often too high for a low-bandwidth access link. It seems unlikely that these (primarily economic) factors are likely to change any time soon. There may be multiple paths to a location (e.g., using wireless), but access links will remain less reliable than carrier-grade backbone links.

A scarcity of AS links at the edges. Maximum path disjointness is primarily determined at the edges of the network. For small-degree multi-homed AS pairs, the total number of disjoint paths between them (calculated using min-vertex-cut) is nearly always determined by the minimum degree of the two nodes in the pair. This confirms intuition that there is significantly more AS link diversity in the core of the network than at the edges.

Single points of failure can limit availability. Paths that share components are susceptible to failures at common locations. If two paths from a source AS S towards a destination D traverse the same upstream provider of D, then a routing failure in that provider AS can bring down both paths from S to D. Completely disjoint paths can eliminate single points of failure, assuming the paths do not experience correlated failures.

If all links are created equal, then choosing between completely disjoint paths maximally improves availability. In practice, of course, some links are better than others, but path disjointness is still a reasonable proxy for failure disjointness. As such, it is a common goal of route control systems and multi-homing techniques [36, 1] and of proposals for improved inter-domain routing [24, 46]. Recent work on consensus routing shows that using a "most-disjoint" backup path for transient routing connectivity is robust to individual stub link failures [21]. In addition, our real-world measurement results (§5) suggest that having disjoint paths does significantly improve availability.

For simplicity and to keep the number of paths manageable, in this portion of the analysis we target AS path disjointness instead of router or link disjointness. An AS disjoint path is router and link disjoint, but is also more likely to be *administratively* disjoint, and hence subject to fewer sources of correlated failures while still able to tolerate individual router and link failures. Of course, exceptions exist that still affect many ASes, and we do not claim that AS disjointness would provide perfect resilience to all of them. Examples include prefix hijacking, BGP errors such as the well-known AS7007 incident [10],

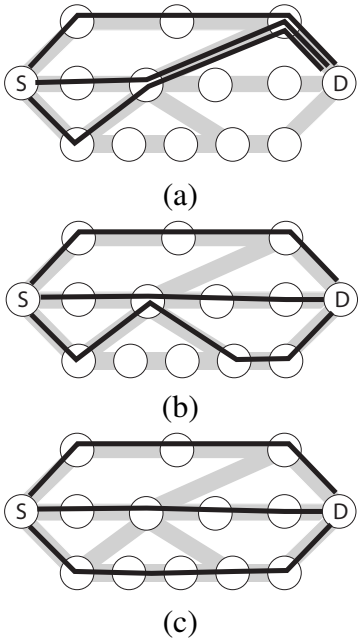


Figure 1: Example AS Topology. (a) Paths using *First Hop Choice* collide at an ingress. (b) Two paths using *First+Last Hop Choice* collide in the middle. (c) *Full Path Choice* obtains maximum path disjointness.

or IP links that traverse the same physical conduit [12]. While these remaining sources of correlated failures can reduce the benefits of AS disjoint paths, it is the current best proxy for failure disjointness.

3.2 Analysis Method

We define three AS path construction mechanisms:

- *First Hop Choice*: The source AS chooses only through which outbound AS to send a packet, using the shortest policy-compliant path from the source's neighbors to the destination.
- *First+Last Hop Choice*: The source specifies the first and last hop AS link to route through, using the shortest policy-compliant path through the core.
- *Full Path Choice*: The source can construct arbitrary AS paths to reach the destination.

First Hop Choice is similar to today's "intelligent route control" systems available to multi-homed organizations [1]; it is the current state of the art in deployed path selection mechanisms.

First+Last Hop Choice increases the amount of path choice over *First Hop Choice* by allowing a multi-homed source communicating with a multi-homed destination to choose through which immediate provider to reach that destination, but without controlling the path between the source's providers and the destination's providers.

Full Path Choice can achieve the maximum number of disjoint paths between the source and destination; knowing the upper bound helps us understand the effectiveness of the first two models and how much room is left for improvement.

Calculating Disjointness: *Full Path Choice* will always obtain the maximum number of AS disjoint paths, which is the min-vertex-cut between the source and destination. For *First Hop Choice* and

First+Last Hop Choice, we compute the **min-vertex-cut** of the *subset* of possible paths constructed by each mechanism, and compare this number to the maximum.

Figure 1 shows an example topology (ignoring relationships for simplicity), and the shortest possible paths chosen under each mechanism. Under *First Hop Choice*, all three paths collide at the upper right node, as the shortest path from each of the source's upstreams go through the same entrance to the destination, yielding only 1 effective disjoint path.

Under *First+Last Hop Choice*, the source can choose a matching between the first hop AS and the last hop AS; Figure 1 shows the best possible matching. The top two paths are disjoint, but the bottom path shares an AS with the middle path, so *First+Last Hop Choice* provides 2 completely disjoint paths.

Finally, *Full Path Choice* can obtain 3 disjoint paths, even if it requires using longer paths. In this example, *First Hop Choice* and *First+Last Hop Choice* fail to obtain as many disjoint paths as *Full Path Choice*.

Data Sources and Inputs: We measure the number of AS disjoint paths under each path model for all pairs of ASes that have between two and five upstream providers in a February 2007 annotated AS relationship graph [11], representing nearly 85 million AS pairs; we choose these small-degree ASes as they are most likely to be multi-homing for reliability.[1] In our analysis, we use shortest policy-compliant paths to emulate the predominant BGP route selection metric.

Verifying with Traceroutes: To verify the accuracy of our analysis on inferred topologies, we also compute the number of AS disjoint paths between 1.5 million AS pairs using iPlane traceroutes [29]. Traceroute data provides information about the real paths that packets take through the network, and helps account for biases in static AS topology measurements. We map traceroute paths to AS paths and calculate disjointness under the path models described above.

3.3 Analytical Results

Figure 2 plots the fraction of AS pairs that achieve maximum AS path disjointness under both *First+Last Hop Choice* and *First Hop Choice* path models. For example, of all 85 million AS pairs considered in the AS topology, 92% of pairs achieve as many AS disjoint paths using *First+Last Hop Choice* as they can with the optimal *Full Path Choice* mechanism. In contrast, *First Hop Choice* provides the maximum number of disjoint paths for far fewer AS pairs in both inferred topology and traceroute measurements.

First+Last Hop Choice and short path lengths ensure disjointness: *First+Last Hop Choice* performs well because optimal AS disjointness is determined primarily near the edges. The core has plenty of AS link diversity, so it is important for paths to use all the available upstreams of both the source and destination. Most of the AS paths we consider consist of between 5 and 6 ASes. For paths of length 5, *First+Last Hop Choice* specifies every AS on the path except the middle (third) AS, and the paths rarely route through the exact same middle AS [43]. Recent work has shown that the diameter of the AS graph has been shrinking over time, suggesting that these results are likely to hold into the future [27]. Complex mechanisms that express more paths between the edges generally cannot produce more disjoint paths, since *First+Last Hop Choice* is almost always an optimal strategy.

Unequal Path Lengths Reduce Disjointness Using *First Hop Choice*: The number of disjoint paths using *First Hop Choice* is relatively low. This is due in part to biases introduced by unequal AS path lengths for paths between a multi-homed source and destination. As shown in Figure 1(a), one of the

[1] We performed our analysis on several AS graphs, differing in both date of inference (May 2005, Feb 2007, May 2007) as well as inference algorithm (CAIDA [11], Gao [17], He [20]), and found that our results were very similar for all graphs [43].

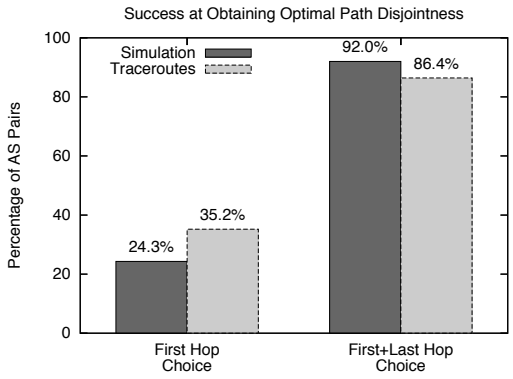


Figure 2: *First+Last Hop Choice* **obtains a maximum number of AS disjoint paths for most AS pairs: values shown for analysis on Feb 2007 CAIDA graph and May 2007 iPlane traceroutes.**

destination's providers is closer to all three of the source's providers, so only that one provider will be used to route traffic between the source and destination.

To quantify the degree to which unequal path lengths play a role, we compute the path lengths between each upstream neighbor and the closest Tier-1 for all 2-multi-homed ASes in our study. Out of the approximately 9500 2-multi-homed ASes in our measurement, we find that half of these ASes have upstream providers with different distances to the closest Tier-1. Most paths collide at the Tier-1s, reaffirming previous results on multi-homing and path diversity [26].

The discrepancy between traceroute and inferred topology results suggests that the topology-based analysis slightly overestimates the effectiveness of *First+Last Hop Choice* and underestimates the disjointness provided by *First Hop Choice*. The overestimate is likely due to inaccuracies in AS topology inference and a known overestimate of shortest-path policy prevalence [39], while the underestimate can be attributed to ASes that balance traffic using path padding.

However, even if ASes use path-padding to balance their path lengths to the core, there is at most a 50% chance that a multi-homed source will be exposed to paths going through all of a destination's upstream providers. In the best case, each of the source's providers will observe paths of equal length through all of the destination's upstream providers. For the source to obtain completely disjoint paths, its own providers must independently forward a path traversing a unique upstream AS. The likelihood of the source obtaining paths through all of the destination's providers is

$$P(m,n) \quad = \quad \sum_{i=0}^{n} (-1)^i \binom{n}{2} (1 - \frac{i}{n})^m$$

where *m*,*n* are the number of providers of the source and destination, respectively.

This probability is upper-bounded by 50% for $m,n \geq 2$, and drops rapidly with increased multi-homing. Thus, *First Hop Choice* is unlikely to obtain a maximum number of disjoint paths for more than 50% of AS pairs without explicit coordination between the organization's providers.

4 Active Measurement Method

Our analysis of path disjointness in the AS topology shows that having choice of both the first and last AS hops can almost always provide disjointness. In contrast, intelligent multi-homing (*First Hop Choice*) rarely obtains a maximum number of disjoint paths.

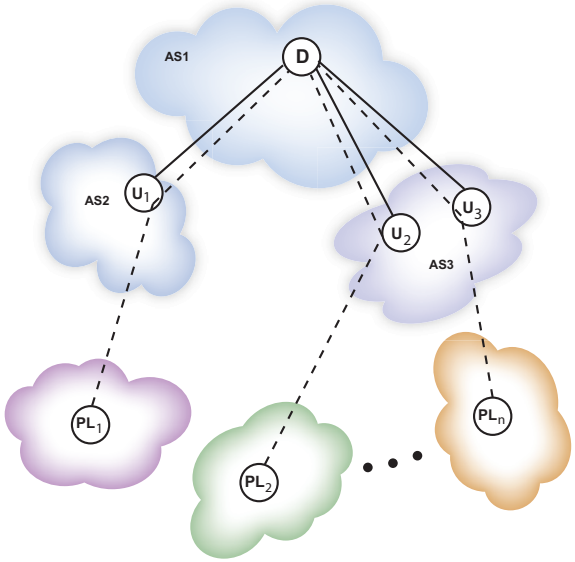


Figure 3: Traceroute paths (denoted by dashed lines) from PL_i nodes to BGP atom destination D traverse immediate upstream routers U_1, U_2, and U_3. We also check that U_i to D latency is $<$ 2.5ms to ensure D and U_i nodes are roughly in the same location to find "locally multi-homed" destinations.

We therefore conducted an active measurement study to understand whether (and to what degree) AS path disjointness translates into improved failure resilience. While our previous analysis showed AS topological insights for improved availability, our active measurement study attempts to quantify the *real-world end-to-end availability benefits* that AS disjointness can provide.

4.1 Node Selection

To compare the effectiveness of multi-homing mechanisms using active measurements, we require multi-homed sources that we control, and destinations that are verifiably multi-homed. Below, we describe our methods for obtaining probe sources and probe destinations that are AS multi-homed to evaluate *First+Last Hop Choice*.

4.1.1 Multi-homed Probe Destinations

To find AS multi-homed destinations that respond to probes, we look for destinations in the Internet with the following characteristics:

1. The destination must have between two and five upstream providers.
2. We must be able to identify the destination's set of immediate upstream routers, which are addressed out of the provider's address space.
3. Every upstream router must be geographically close to the destination.
4. The routers must respond to probes from our sources.

The first requirement is instituted to evaluate the same type of ASes measured in our analysis in §3.2. The second requirement ensures that the destinations we choose allow us to evaluate the *First+Last Hop Choice* path model described in §3. The third requirement ensures that the destinations we choose are "locally multi-homed destinations": multi-homed stub ASes seeking higher availability and perfor-

(lat,long)	Hostnames	IP Addresses	ASNs
(37.870,-122.268)	planetlab5.millennium.berkeley.edu, planet2.berkeley.intel-research.net	169.229.50.14, 12.46.129.22	25, 7018
(42.388,-72.524)	planetlab2.engr.uconn.edu, planetlab1.cs.dartmouth.edu, planetlab2.cs.umass.edu	137.99.11.87, 129.170.214.191, 128.119.247.211	13796, 10755, 1249
(44.040,-123.060)	planetlab2.cs.uoregon.edu, planetlab2.een.orst.edu	128.223.8.112, 128.193.33.8	3582, 4201
(40.445,-79.948)	planet1.pittsburgh.intel-research.net, planetlab-2.cmcl.cs.cmu.edu	12.108.127.136, 128.2.223.64	7018, 9
(33.970,-118.250)	planetlab1.postel.org, planetlab1.cs.ucla.edu, planetlab3.nbgisp.com	206.117.37.4, 131.179.112.70, 63.64.153.84	4, 52, 18473
(35.750,139.500)	planet0.jaist.ac.jp, planetlab-04.naist.jp, pub2-s.ane.cmc.osaka-u.ac.jp	150.65.32.66, 163.221.11.74, 133.1.74.163	17932, 2500, 4730
(25.020,121.370)	planetlab1.iis.sinica.edu.tw, planetlab2.ntu.nodes.planet-lab.org	140.109.17.180, 140.112.107.82	9264, 17716
(42.378,-71.116)	planetlab1.csail.mit.edu, lefthand.eecs.harvard.edu, planetlab-02.bu.edu	128.31.1.11, 140.247.60.123, 204.8.155.227	3, 11, 111
(38.910,-77.020)	node1.lbnl.nodes.planet-lab.org, planetlab2.georgetown.edu	198.128.56.11, 141.161.20.33	16, 11318
(40.765,-111.844)	planetlab2.flux.utah.edu, planetlab2.byu.edu	155.98.35.3, 128.187.223.212	17055, 6510
(40.489,-74.408)	planet2.att.nodes.planet-lab.org, planetlab1.rutgers.edu	204.178.4.164 ,165.230.49.114	6431, 46
(55.750,37.600)	pl1.grid.kiae.ru, plab-2.sinp.msu.ru	144.206.66.56, 213.131.1.102	6801, 12925

Figure 4: List of PlanetLab probe sources grouped by approximate location. Latitude and longitude values obtained through IP-to-LatLong translation from CoralCDN database. Each group of probe sources emulates a multi-homed source with the PlanetLab nodes as the source's providers.

mance will choose multiple upstream providers located in the same city, since it is more expensive to obtain a direct connection to an upstream router in a different city. ASes with the ability to peer with geographically distant ASes are likely to be more highly available than the ones we study. Finally, the fourth requirement limits the destination set to those we can probe for availability and are not filtering or significantly rate-limiting our probe traffic.

Finding Immediate Upstream Routers: Using the iPlane traceroute dataset [29], we identify those BGP atom destinations that reside in ASes with degree between 2 and 5. For each destination, we merge the IP traceroute path from each iPlane source to the destination and generate a reverse path tree, where the root node is the destination and the leaves are the iPlane traceroute sources (Figure 3).

Starting from the root (D), we search among all adjacent nodes to find each immediate upstream router in a different AS than that of the destination (routers U_1, U_2, and U_3 in Figure 3). Each BGP atom destination for which we identify upstream routers in at least two of its providers qualify as potential destination candidates, since PlanetLab traceroute sources are able to contact the destination AS (AS1) through different upstream ASes (AS2, AS3).

Verifying Proximity: To verify that the destinations are locally multi-homed, we use `ttl`-limited traceroute probes from PlanetLab nodes to more rigorously identify the latency between the destination and the set of immediate upstream routers. Once we identify that both the upstream router and the destination respond to ICMP probes, we then send 5 probes to each router, taking the minimum reported delay as the value most indicative of distance, also ensuring that the ICMP responses are not being rate-limited. We then subtract the delay to the destination from the delay to the upstream router to estimate the round-trip-time between the upstream router and the destination.

If the delay between the upstream and the destination is below 5ms (a one-way delay of 2.5ms), we consider the pair to be sufficiently close together. We identified 2943 multi-homed destinations for our study.

4.1.2 Probe Sources and Destination Assignment

Emulating multi-homed sources: We emulate a multi-homed source using two or more PlanetLab nodes located in the same city, a technique borrowed from Akella et al. [1]. For example, we use nodes at Intel Research Pittsburgh and Carnegie Mellon University as a pair emulating one multi-homed source.

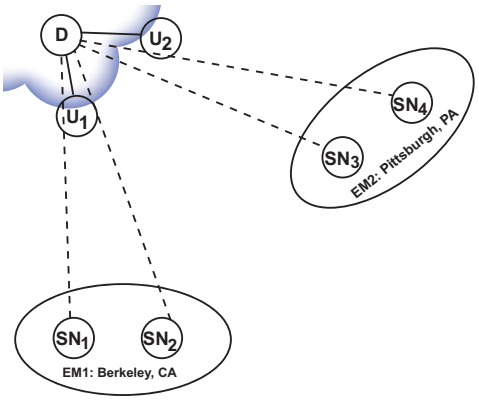


Figure 5: Multi-homed destination *D* is probed by 2 emulated multi-homed nodes consisting of 4 source nodes.

We identify these sources automatically by using the CoralCDN IP geolocation database to map PlanetLab nodes to a latitude and longitude, and then grouping nodes if the distance between them is 5 miles or less.

We call each node a *source node (SN)*, and a set of source nodes in the same city an *emulated multi-homed node*. We obtained 12 emulated multi-homed nodes consisting of 28 source nodes.[2] 8 of our emulated multi-homed nodes emulate a 2-multi-homed source and 4 of them emulate a 3-multi-homed source. Each source node runs a Scriptroute daemon [40] for probing.

Assigning destinations to emulated multi-homed nodes: We enlist 4 emulated multi-homed nodes to probe a given destination, choosing those nodes with paths through several upstream entrances available to the destination (Figure 5).[3] Because PlanetLab nodes sometimes fail during availability probing, we assign 2 emulated multi-homed nodes per upstream (i.e. 4 PlanetLab nodes probe a destination for every AS upstream). Thus, a destination with two AS upstreams is probed by 4 emulated multi-homed nodes, consisting of 8 to 10 source nodes.

Measurement statistics: Each source node probes at most 25 destinations simultaneously, a limit instituted to bound the probing rate of any given node to fall within acceptable use and to ensure that probes are not lost due to PlanetLab network slice scheduling variation.[4] Given this restriction, we probed 110 of the 2943 potential destinations from 12 emulated multi-homed nodes during the week of April 11–18, 2008. After removing destinations that were available less than 90% of the measurement period and paths that provided less than 5 days worth of probe results, we obtain path data for 629 (source,destination) pairs.

4.2 Probing

Probe timing: Our probe timing is based on SOSR's probing algorithm for measuring availability [19]. Each source node sends ICMP probes to a destination every 20 seconds in a normal state. When a destination does not respond within 3 seconds, the path transitions into a loss state. In a loss state,

[2]We originally obtained 18 emulated multi-homed nodes consisting of 43 source nodes, but many of the nodes failed for significant periods of time during probing, so we removed them.

[3]All destinations we consider here are 2-multi-homed destinations, though they may have several IP upstreams.

[4]When we probed 50 destinations every 10-20 seconds, `libpcap` often could not keep up with the rate of outgoing packets.

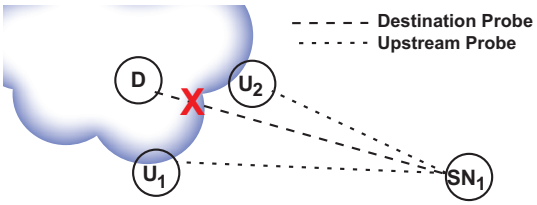


Figure 6: When $SN1$ cannot reach D, it probes D's upstreams, U_1 and U_2, and records reachability and latency. By probing all upstreams, even those the default path does not traverse, we can potentially discover whether the destination could have been reached through another inbound path to the destination.

the probe server experiencing the probe loss probes the destination every 10 seconds. In addition, the server probes each upstream router and records reachability (Figure 6). Upon receiving 5 consecutive successful probe responses from the destination, the path returns to the normal state.

4.2.1 Definition of Failure

We define a path failure as path unavailability lasting more than 30 seconds, or in other words, when three consecutive probes to a destination do not receive a response. This differs from the SOSR definition of a failure: In our study, 3 probe losses followed by 1 probe success and another 3 probe losses are treated as two separate 30 second failures, whereas SOSR would classify this as a 70 second failure. Thus, our availability numbers represent an absolute uptime, rather than an end-user perceived uptime.

While our probe granularity does not capture transient failures lasting less than 30 seconds, we chose this number to help differentiate congestion-related losses from real path unavailability: congestion may explain 1 or even 2 consecutive probe losses, but the probability of losing three consecutive probes (spaced 10 seconds apart) due to congestion is likely to be low: on a path experiencing 1% packet loss, three consecutive losses would occur one out of every million probes (assuming losses 10 seconds apart are relatively independent). Because each path sees fewer than 100,000 probes over the course of one week, false failures introduced due to congestion are unlikely.

4.2.2 Upstream Tracking

The multi-homed destinations in our study do not have upstream routers that support packet deflection. Evaluating *First+Last Hop Choice* requires knowledge of whether the destination can be contacted through any upstream routers: we must be able to determine the upstream used for probes that successfully reach a destination. This is done by sending `ttl`-limited probes and registering responses with a destination→upstreamList database.

Tracking upstreams using time-to-live: To understand which upstream a given probe is going through, each normal probe to the destination has its `ttl` set to the current number of hops to the destination. When a path gets longer, the probe we receive has a source address that is different from the destination, so the upstream has potentially changed. We then perform a traceroute to the destination to identify the new upstream and the new number of hops to the destination.

Paths can also get shorter or use a different upstream, and sending `ttl`-limited probes set to expire at the destination will not capture these cases. Thus, we also periodically (every 5 minutes to prevent probing overload) send a `ttl`-limited probe with the `ttl` set to terminate at the expected upstream. If

the response we receive is not from the upstream currently listed, we update the current upstream value to the one just discovered.

Upstream router database: We seed each server with a list of destinations and the associated upstreams discovered during node selection. The upstream list we provide is often incomplete because later probing identifies new upstreams (e.g. introduced due to path changes, fail-over links). We therefore maintain a database of destination→upstreamList entries that are updated to reflect which immediate upstream routers a source node has routed through to reach a destination.

When a source node experiences a path failure, it queries the database to get an updated list of upstreams to probe. When a source node observes a new upstream, it registers the upstream in the database.

4.3 Measurement Issues

Clock skew: In our data analysis, we compare probes from different sources. A large clock skew on the PlanetLab nodes (e.g., greater than 10 seconds) would shift probes to different measurement bins, causing error [33].

To prevent such errors, we probed every source node every minute to obtain their NTP or ICMP timestamp response, and then calculated the drift over time. After removing one problem node from our experiment, no source node drifted further than 5 seconds from any other node in the experiment, ensuring that their probes fell into the correct interval.

Lack of real multi-homed sources: Our probe sources do not perfectly represent a multi-homed source. Our construction may underestimate the impact of correlated failures, e.g. due to power outages at the source or to failures at a local exchange point where logically disjoint paths physically meet. However, as our focus is on destination-side failures, we are mostly concerned with visibility into failures at the core and near the destination.

5 Active Measurement Results

This section presents our active measurement results. We describe the path availability provided by 5 path models, including *First Hop Choice* and *First+Last Hop Choice*. In addition, we analyze the distribution of length of failures observed in our study to provide intuition for the kinds of failures that each path model experiences or avoids. Lastly, we briefly discuss the latency characteristics of the paths measured in the study.

5.1 Availability

Path Models: We consider five path models in our study:

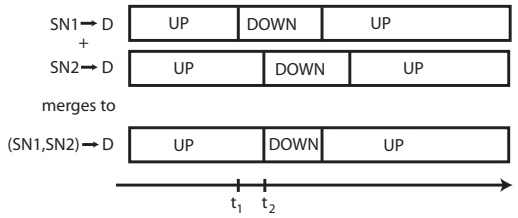
- *Single Path* - Single Path routing: uses only the default path to the destination.
- *First Hop Choice* - Intelligent multi-homing: can use the default path from any upstream provider to the destination.
- *Last Hop Choice* - Uses the default path to the destination, or the path from the source to destination through any of the destination's upstream providers.

- *First+Last Hop Choice* - Uses intelligent multi-homing, but also allows choice of deflecting through any of the destination's upstreams.
- *DestReach* - The upper bound on availability: how often the destinations are available.

Calculating Availability:

For *Single Path*, we calculate the availability percentage by dividing the total uptime by the total measurement period for the path from a source node $SN1$ to a destination D, denoted as $SN1-D$.

For *First Hop Choice*, we merge the availability for $SN1-D$ and $SN2-D$, where $SN1$ and $SN2$ form an emulated multi-homed source. As shown below, if the path $SN1-D$ was down from t_1 to t_2 but $SN2-D$ was available during that time, we consider the emulated multi-homed node with upstream providers $SN1$ and $SN2$ to have 100% availability to D during this time period.



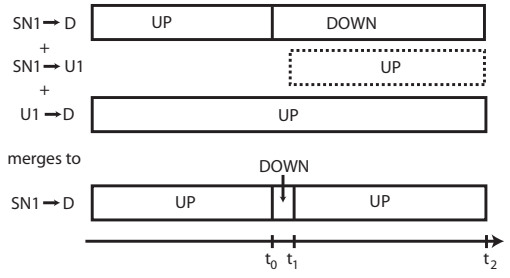
For *Last Hop Choice* and *First+Last Hop Choice*, we use the paths provided by *Single Path* and *First Hop Choice*, but also consider the paths going through the upstreams to the destination. Because we cannot deflect packets through the last hop routers of the real destinations, we use the probing data from other PlanetLab nodes to observe the availability of the link between all last hop routers and the destination.

For a given source,destination pair (S,D) with upstream set U during time-interval t, we can calculate the availability through all upstreams as:

$$Avail(S,D,U,t) = Avail(S,D,t) \mid (\forall i.\ (Avail(S,U_i,t)\ \&\ Avail(U_i,D,t))),$$

where $Avail(a,b,t) = 1$ if a could contact b during time interval t, and 0 otherwise.

We illustrate this in the figure below, where a source node $SN1$ is unable to reach the destination D during a time interval $[t_0, t_2]$ because the upstream it traverses has failed. However, another source node has successfully probed the destination during this time interval using a path through upstream U_1. Thus, we have proof that the link from U_1 to D is working. During time interval $[t_1, t_2]$, $SN1$ probes all of D's upstreams, including U_1. If the path from $SN1$ to U_1 is working and the path from U_1 to D is working, we obtain a piecewise proof that the path from $SN1$ to D is available through an alternative upstream U_1 during interval $[t_1, t_2]$. This allows us to measure the availability benefits of last hop choice without having control of packet deflection at the last hop.



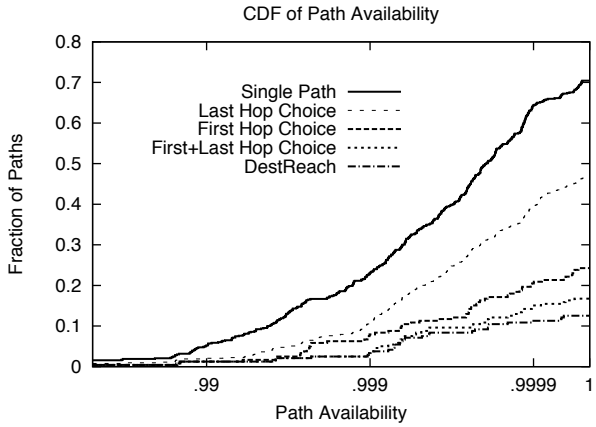


Figure 7: CDF of path availability under different path construction schemes.

DestReach, or *destination reachability*, measures how often the destinations were observably available from any source at all. Because we assign eight to ten source nodes to every destination, many other sources can potentially reach the destination when one emulated source node cannot.

5.1.1 Availability Results

Figure 7 plots the CDF of availability using each path model. The x-axis represents the path availability as a fraction of time that the path worked during the week-long measurement period, while the y-axis represents the fraction of paths that achieved availability less than the corresponding point on the x-axis. For example, the height where the lines meet the right edge of the graph indicates what fraction of paths achieved less than full (100%) availability.

Using *Single Path* routing, 70% of paths experienced less than 100% availability. Single-homed sources contacting multi-homed destinations therefore observe the destinations as being often unavailable, despite the destination's best efforts to multi-home.

By allowing a source to deflect through any of a destination's upstreams (*Last Hop Choice*), only 50% of paths experienced less than 100% availability: 20% of paths were able to eliminate all failures experienced under *Single Path* by deflecting through a last hop router.

Multi-homed sources that can select which upstream to use (*First Hop Choice*) experience higher availability: only 25% of paths are available less than 100%. Since a majority of paths under *First Hop Choice* are likely to collide near the destination, this improvement implies resilience to failures near the source or in the core of the Internet.

Using *First+Last Hop Choice*, 84% of paths exhibit 100% availability: about 10% of paths that experienced failures when they could only select an outbound route were now failure free.

The availability of each of the path models above is upper-bounded by destination reachability (*DestReach*), i.e. path failures where the actual destination was unavailable (e.g. router reboots, power outages). As shown in Figure 7, 12% of the destinations we consider were unavailable during the measurement period for some amount of time. Only 4% of paths experience a failure where *First+Last Hop Choice* is unable to reach the destination when the destination was provably available. We discuss potential explanations for these 4% of paths in the next section.

In summary, we find that providing last hop choice can significantly improve availability for both

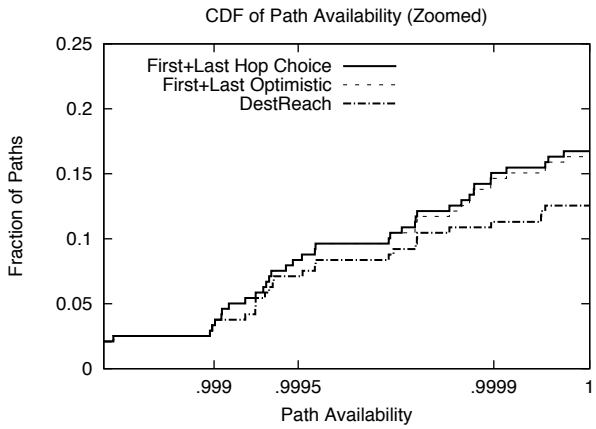


Figure 8: CDF of path availability, zoomed in to analyze differences between *First+Last Hop Choice*, *First+Last Optimistic*, and *DestReach*. When *First+Last Hop Choice* fails to reach the destination that is available, the emulated multi-homed node cannot reach any of the destination's upstreams.

single-homed and multi-homed sources: 10% of paths that experienced a failure under *First Hop Choice* were made fully available when given the ability to choose the first and last AS hops. Moreover, the paths provided by *First+Last Hop Choice* are almost always as available as the destinations themselves, leaving little room for further availability improvements.

5.1.2 Understanding the Gap Between *First+Last Hop Choice* and *DestReach*

We consider an additional path model: *First+Last Optimistic*, where we simply require the destination's upstreams be reachable when *First+Last Hop Choice* is unable to reach the destination. As shown in Figure 8, this provides almost no improvement over *First+Last Hop Choice*, indicating that when *First+Last Hop Choice* fails, none of the source nodes in the emulated multi-homed node could reach any of the upstreams of the destination.

There are two potential explanations for the gap between *First+Last Hop Choice* and *DestReach*:

1. Correlated failures: neither of the source nodes could reach the destination or the upstreams, but another source node in another city could.
2. Probing an incomplete set of upstreams: during a failure, a source node *SN*1 obtains a list of the upstreams to a destination by querying our upstream-tracking database. For short failures, however, another source node may be successfully reaching the destination through a backup upstream router that is only visible during this short failure. Thus, *SN*1 does not probe the backup router, and hence we do not obtain enough information to know whether this failure could be avoided.

Given our data, we are unable to completely distinguish between correlated failures and an incomplete upstream knowledge. Since upstreams are only probed every five minutes, we will not know of the existence of a temporary backup router during short failures.

However, Figure 8 shows that paths with less than 99.9% availability were all down specifically because the destination itself was down. A downtime of 5 minutes in one week represents a path availability of 99.65%, so all failures observed by *First+Last Hop Choice* but avoided by *DestReach* are shorter than 5 minutes. This leads us to believe that the gap is more likely explained by an incomplete set of upstream routers than by correlated failures.

	Single Path	*First Hop Choice*	*First+Last Hop Choice*	*DestReach*
Median Failure Duration (s)	47.011	110.316	110.316	104.316
99%-ile Failure Duration (s)	1944.605	5132.227	5224.732	5229.299
Longest Failure Observed (s)	39401.801	27187.302	27187.302	27187.302
Avg. Total Failures/Path	9.348	1.944	1.500	1.131

Figure 9: Failure characteristics for *Single Path*, *First Hop Choice*, *First+Last Hop Choice*, and *DestReach*.

5.2 Failure Characteristics

To understand the kinds of failures experienced by each path model, we present failure duration numbers for four path models in Figure 9.

The median failure duration under *Single Path* is 47 seconds: most failures observed by *Single Path* are relatively short. On the other hand, *Single Path* also observes a failure that is longer than those experienced by other path models.

For *First Hop Choice*, *First+Last Hop Choice*, and *DestReach*, the failure distribution is nearly identical. A closer inspection of the actual failures observed shows that all failures longer than 600 seconds were due to destination failures. Thus, ***First+Last Hop Choice* only improves upon *First Hop Choice* for failures shorter than 10 minutes**.

Under *Single Path*, one path experienced a failure for half a day that could not be explained by a destination failure. In this case, the routing infrastructure was unable to make that path available. However, another node in the same city was able to reach that destination, allowing *First Hop Choice* to circumvent the failure.

To more clearly illustrate where *First+Last Hop Choice* improves upon *First Hop Choice*, we also plot the failure duration CDF on an absolute scale in Figure 10, where the y-axis represents the cumulative absolute downtime rather than over a fraction of failures (note the log-log axes). We observe the following:

- Single-path routing experiences a significantly larger number of short failures than all multi-path models.
- *First+Last Hop Choice* eliminates several short failures that *First Hop Choice* experiences. Since many paths observe only a few failures over the entire measurement period, each failure eliminated provides a significant improvement in availability.
- The destinations we consider were rarely down for more than 500 seconds, and only a few destination failures lasted longer than 2000 seconds. However, these long destination failures significantly affect the total cumulative downtime experienced by all models.

We note that because the *number* of paths considered for *Single Path* and *Last Hop Choice* is twice that of the multi-path models, the cumulative downtime should not be directly compared.

5.3 Latency

While their major goal is often to improve availability, alternative path mechanisms also care about performance: if the default path fails, the alternative path should perform somewhat similarly to reduce the impact of a failure on a particular path. Because we do not control the multi-homed destinations

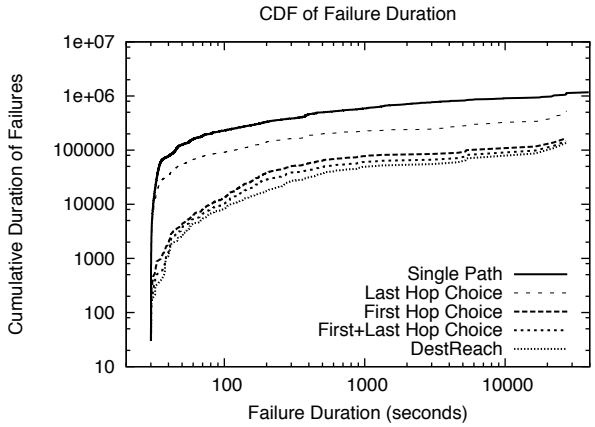


Figure 10: CDF of failure duration on an absolute scale.

in this study, we analyze latency improvements but not bandwidth improvements, the latter being much more difficult to measure at the scale of our study.

In our analysis, we break time into 10 second intervals and aggregate all probe responses for a given source, destination pair during this interval. For *First Hop Choice* and *First+Last Hop Choice*, there exist several latency choices (one for every path), and the value chosen for the interval depends on the path selection algorithm used. We consider two path selection models in this paper:

1. *Omniscient*: the source can choose the best path during a given time interval based on the results.
2. *Predictive*: the path with the best latency during interval i is the path whose latency will be used for interval $i+1$.

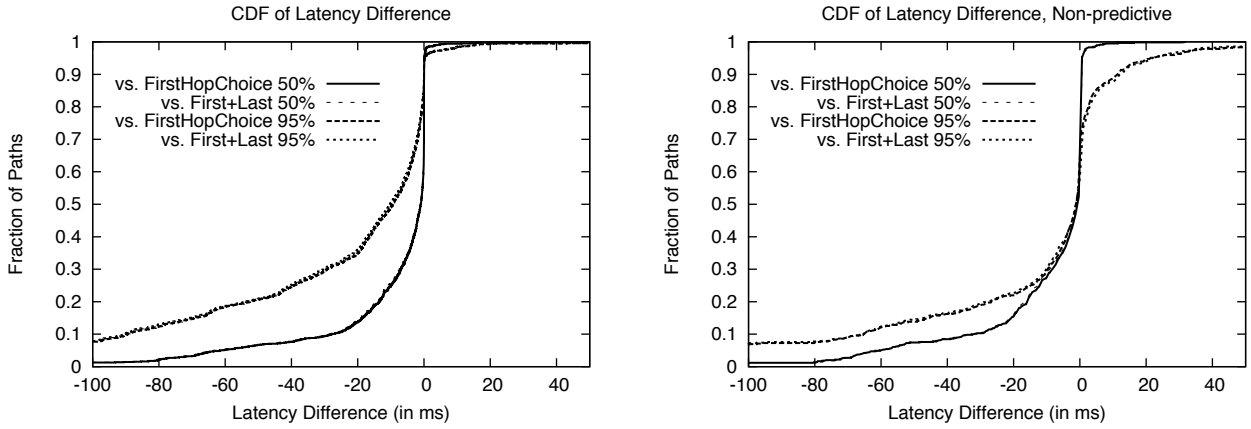
The omniscient model represents an upper-bound on the performance benefits of *First Hop Choice* and *First+Last Hop Choice*. The predictive model is a straw-man path selection scheme that is more susceptible to noise. More complex path selection algorithms exist (e.g. [4]), but the performance of other models should fall somewhere between the two strategies we study.

5.3.1 Latency Results

In this section, we confirm previous results showing that alternative path selection can provide improvements to latency. Furthermore, we find that *First+Last Hop Choice* does not provide significantly better performance in comparison to *First Hop Choice*.

Using *Omniscient* path selection: As shown in Figure 11(a), the alternative paths offered by multi-homing can significantly improve path latency compared to single-path routing under an omniscient path selection scheme. **For 25% of paths, alternative paths improve median latency by at least 10ms**. In most cases, one of the upstreams has a better path to the given destination, and as a result, the average latency level is dominated by the quality of the better path. This reaffirms previous results that any path choice can greatly improve latency. The improvement at the 95%-ile level suggests that multi-homing provides some resilience to single path routing anomalies: each path consists of 25000 probe intervals, during which severe routing anomalies occur often enough to create a large performance gap between paths for at least 1250 probe intervals.

First+Last Hop Choice does not significantly improve path latency compared to multi-homing at any percentile. There are two explanations for this lack of improvement:



(a) Omniscient Selection (b) Predictive Selection

Figure 11: CDF of median and 95%-ile path latency difference compared to *Single Path*.

- One source's path to the destination is almost always better than the other, and thus the benefits of *First+Last Hop Choice* are not seen at the percentile levels shown.
- The paths provided by *First+Last Hop Choice* are roughly the same in terms of geographic distance traveled, since the alternative deflection point exists in the same city as the destination.

These observations reaffirm hypotheses from previous work that much of the latency improvements of alternative path routing come from avoiding a bad path, not finding the best one [19, 28, 38].

Under *Predictive* path selection: Figure 11(b) shows that *Predictive* path selection for *First Hop Choice* and *First+Last Hop Choice* does not provide better paths as often as omniscient selection, although the overall conclusions are roughly the same.

In summary, omniscient path selection provides an upper bound on the achievable latency benefits of both *First Hop Choice* and *First+Last Hop Choice*, but the upper bound does not differ significantly from predictive selection models. These results help to explain the benefits observed previously [5], and show that while simple multi-homing can significantly improve latency, further flexibility does not provide additional benefits.

6 Discussion

6.1 Implementing *First+Last Hop Choice*

The focus of this work is on the methods and measuring the benefits of multi-homing without regard to implementation technique. Given that *First+Last Hop Choice* appears to provide significant availability benefits over multi-homing, however, it is worth discussing these issues briefly. We highlight *one* possible way to implement *First+Last Hop Choice* in today's architecture.

First+Last Hop Choice requires that a multi-homed enterprise advertise last-hop reachability. Two additional services are needed beyond what the Internet currently provides to implement this technique: last-hop packet redirection and an out-of-band dissemination system allowing discovery of IP-prefix→last-hop addresses.

Redirection using IP-in-IP encapsulation: A multi-homed source must be able to choose a path to a destination through an alternative upstream router. Encapsulation has been proposed as one way to provide this choice [44, 35], and is supported at line-speed in modern routers [16]. These last-hop routers must then simply determine whether the packet is destined for one of their supported destinations. As a result, this redirection can be implemented effectively in today's architecture.

Disseminating last hop mappings: A multi-homed source must be able to obtain the list of last-hop routers for a given destination in an out-of-band fashion. We assume that a multi-homed AS will want to provide increased availability to all nodes within the AS, and so an AS need only publish mappings for destinations at the IP-prefix granularity. Thus, mappings represent an IP-prefix→router-list association rather than a per-destination mapping.

This mapping granularity admits several mechanisms for disseminating the mappings between a destination IP prefix and a list of last-hop routers. Fortunately, these mechanisms can be simple, because the mappings need not be kept aggressively up-to-date for the system to function well. Stale mappings do not reduce availability below that provided by default BGP routing, because the source can always elect to use the default path. Moreover, mappings reflect a business agreement between a customer and a provider, and thus should change infrequently.

To quantify the size of the global mapping state, we estimate that a mapping between a destination prefix and a set of last-hop routers will require no more than 30 bytes per prefix: 32+8 bits for the prefix and mask for a set of destinations, and 32+8 bits for the last-hop router IP and some additional data for traffic engineering/policy uses, and we restrict each prefix to five last-hop routers (further multi-homing is unlikely to significantly improve availability). The size of the table scales linearly with the number of prefixes in the table, and current backbone routing tables contain roughly 235,000 prefixes. Casually rounding up the number of prefixes to the nearest million to compensate for routing table growth as a result of increased multi-homing, the full mapping table would contain less than 30MB.

While even this number is quite manageable, disseminating table updates should be much easier. In one typical week (Jul 2–9 2007), fewer than 3000 prefixes were introduced or switched to a different provider. We calculated this number by finding all customer ASes that switched to a new provider AS between successive versions of the CAIDA AS topology. We then counted all prefixes newly or formerly advertised via BGP by those customer ASes. 671 prefixes switched, and approximately 2000 new prefixes were introduced in the later BGP table. Thus, less than 90KB of data needs to be sent each week to keep the IP-prefix→last-hop router list up to date.

6.2 Related Work

Several proposals for source routing provide path choice [6, 45, 23, 48], but much of their benefits come at the cost of scalability and policy restrictions; we refer the reader to [44] for a detailed discussion of multi-path routing architectures. Regardless, because of their enormous flexibility, we consider source routing models as the "gold standard" for obtaining high availability and seek to understand where path choice can most effectively provide as high availability as source routing designs.

Both MIRO and Routing Deflections [44, 46] propose new multi-path routing architectures by specifying how to negotiate and use alternate paths to a destination. In contrast, our work attempts to identify *which* ASes should negotiate to obtain disjoint paths that could be used in a future routing architecture providing a small and effective set of failure disjoint paths to end-hosts.

Tunneling packets across the Internet core has been proposed as a way to scale BGP routing [47]. Our work has much in common with CRIO's tunneling [47], the recent Locator/ID separator (LISP)

proposal [35], and Virtual Peerings [34]. We find it encouraging that similar architectures can simultaneously increase availability (our work) *and* improve scalability and traffic engineering (prior work).

Overlay-based techniques like RON, SOSR, and Detour [3, 19, 38] use overlay nodes to find failure-disjoint paths. Our work identifies the important properties that these overlay routing techniques take advantage of; we believe that our results may help inform designs for future overlay networks by making it more clear where and why they should seek to provide disjoint paths.

Previous work on comparing multi-homing and overlay routing shows that a lack of link diversity near the edges of the network are a major factor reducing the availability benefits from both multi-homing and overlays [2]. Their results particularly applied to multi-homed sources contacting single-homed destinations, while we extend that study to consider multi-homed destinations. Others found that path diversity does not significantly improve end-to-end path delay for a small deployment of academic nodes [41]; we try to measure availability from more nodes and paths and obtain similar path delay results.

Failover-based routing mechanisms [8, 24] attempt to provide immediate failover to backup paths upon detected failures. Several researchers have found benefits to having disjoint paths in simulations [24, 21, 15]. In our work, we attempt to measure the benefits of having disjoint paths using real-world active measurements.

Teixeira et al. [42] found that a traceroute-inferred topology of the Sprint backbone under-counted potential path diversity due to missing backup links. While our study is at the autonomous system level, their results provide a warning that similar un-exposed relationships may lead our AS study to also undercount diversity.

7 Conclusion

This paper helps identify where and how much topological knowledge can most effectively improve Internet availability. We characterize the path disjointness and availability achieved by several multi-homing models. Our analytical AS topology results and active measurement study showed that choosing the first and last AS hops provides AS disjoint paths that can substantially increase availability. These results suggest an effective degree of topology exposure for future Internet architectures that balances availability gains with scalability.

References

[1] Aditya Akella, Bruce Maggs, Srinivasan Seshan, Anees Shaikh, and Ramesh Sitaraman. A measurement-based analysis of multihoming. In *Proc. ACM SIGCOMM*, Karlsruhe, Germany, August 2003.

[2] Aditya Akella, Jeff Pang, Bruce Maggs, Srinivasan Seshan, and Anees Shaikh. A comparison of overlay routing and multihoming route control. In *Proc. ACM SIGCOMM*, Portland, OR, August 2004.

[3] David G. Andersen, Hari Balakrishnan, M. Frans Kaashoek, and Robert Morris. Resilient Overlay Networks. In *Proc. 18th ACM Symposium on Operating Systems Principles (SOSP)*, pages 131–145, Banff, Canada, October 2001.

[4] David G. Andersen, Hari Balakrishnan, M. Frans Kaashoek, and Rohit Rao. Improving Web availability for clients with MONET. In *Proc. 2nd USENIX NSDI*, Boston, MA, May 2005.

[5] David G. Andersen, Alex C. Snoeren, and Hari Balakrishnan. Best-path vs. multi-path overlay routing. In *Proc. ACM SIGCOMM Internet Measurement Conference*, Miami, FL, October 2003.

[6] Katerina Argyraki and David R. Cheriton. Loose source routing as a mechanism for traffic policies. In *ACM SIGCOMM Workshop on Future Directions in Network Architecture*, Portland, OR, September 2004.

[7] Hitesh Ballani, Paul Francis, and Xinyang Zhang. A study of prefix hijacking and interception in the Internet. In *Proc. ACM SIGCOMM*, Kyoto, Japan, August 2007.

[8] O. Bonaventure, C. Filsfils., and P. Francois. Achieving sub-50ms recovery upon bgp peering link failures. In *Proc. CoNEXT*, October 2005.

[9] Matthew Caesar, Nick Feamster, Jennifer Rexford, Aman Shaikh, and Jacobus van der Merwe. Design and implementation of a routing control platform. In *Proc. 2nd USENIX NSDI*, Boston, MA, May 2005.

[10] CNET News.com. Router Glitch Cuts Net Access. `http://news.com.com/2100-1033-279235.html`, April 1997.

[11] Xenofontas Dimitropoulos, Dmitri Krioukov, Marina Fomenkov, Bradley Huffaker, Young Hyun, kc claffy, and George Riley. AS relationships: Inference and validation. *ACM Computer Communications Review*, 37:29–40, 2007.

[12] Sean Donelan. Update: CSX train derailment. `http://www.merit.edu/mail.archives/nanog/2001-07/msg00351.html`.

[13] Patricia Enriquez, Aaron Brown, and David A. Patterson. Lessons from the PSTN for dependable computing. In *Workshop on Self-Healing, Adaptive and Self-Managed Systems*, 2002.

[14] Nick Feamster and Hari Balakrishnan. Detecting BGP Configuration Faults with Static Analysis. In *Proc. 2nd Symposium on Networked Systems Design and Implementation*, Boston, MA, May 2005.

[15] Teng Fei, Shu Tao, Lixin Gao, and Roch Guerin. How to select a good alternate path in large peer-to-peer systems. In *Proc. IEEE INFOCOM*, Barcelona, Spain, March 2006.

[16] Pierre Francois and Olivier Bonaventure. An evaluation of IP-based fast reroute techniques. In *Proc. CoNEXT*, October 2005.

[17] Lixin Gao. On inferring automonous system relationships in the Internet. *IEEE/ACM Transactions on Networking*, 9(6):733–745, December 2001.

[18] Albert Greenberg, Gisli Hjalmtysson, David A. Maltz, Andy Myers, Jennifer Rexford, Geoffrey Xie, Hong Yan, Jibin Zhan, and Hui Zhang. A clean slate 4D approach to network control and management. *ACM Computer Communications Review*, 35(5):41–54, 2005.

[19] Krishna P. Gummadi, Harsha V. Madhyastha, Steven D. Gribble, Henry M. Levy, and David Wetherall. Improving the reliability of Internet paths with one-hop source routing. In *Proc. 6th USENIX OSDI*, San Francisco, CA, December 2004.

[20] Y. He, G. Siganos, M. Faloutsos, and S. V. Krishnamurthy. A systematic framework for unearthing the missing links: Measurements and impact. In *Proc. 4th USENIX NSDI*, Cambridge, MA, April 2007.

[21] John P. John, Ethan Katz-Bassett, Arvind Krishnamurthy, Thomas Anderson, and Arun Venkatarmani. Consensus Routing: The Internet as a Distributed System. In *Proc. 5th USENIX NSDI*, San Francisco, CA, April 2008.

[22] Ethan Katz-Bassett, Harsha V. Madhyastha, John P. John, Arvind Krishnamurthy, Thomas Anderson, and David Wetherall. Studying Black Holes in the Internet with Hubble. In *Proc. 5th USENIX NSDI*, San Francisco, CA, April 2008.

[23] H. Tahilramani Kaur, S. Kalyanaraman, A. Weiss, S. Kanwar, and A. Gandhi. BANANAS: an evolutionary framework for explicit and multipath routing in the internet. In *FDNA '03: Proceedings of the ACM SIGCOMM workshop on Future directions in network architecture*, 2003.

[24] Nate Kushman, Srikanth Kandula, Dina Katabi, and Bruce M. Maggs. R-BGP: Staying connected in a connected world. In *Proc. 4th USENIX NSDI*, Cambridge, MA, April 2007.

[25] Craig Labovitz, Abha Ahuja, and F. Jahanian. Experimental study of Internet stability and wide-area network failures. In *Proc. FTCS*, Madison, WI, June 1999.

[26] Sanghwan Lee, Zhi-Li Zhang, and Srihari Nelakuditi. Exploiting as hierarchy for scalable route selection in multihomed stub networks. In *Proc. Internet Measurement Conference*, Taormina, Italy, October 2004.

[27] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graphs over time: Densification laws, shrinking diameters and possible explanations. In *Proc. 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Chicago, IL, August 2005.

[28] Cristian Lumezanu, Dave Levin, and Neil Spring. PeerWise Discovery and Negotiation of Faster Paths. In *Proc. 6th ACM Workshop on Hot Topics in Networks (Hotnets-VI)*, Atlanta, GA, November 2007.

[29] Harsha V. Madhyastha, Tomas Isdal, Michael Piatek, Colin Dixon, Thomas E. Anderson, Arvind Krishnamurthy, and Arun Venkataramani. iPlane: An information plane for distributed services. In *Proc. 7th USENIX OSDI*, Seattle, WA, November 2006.

[30] Ratul Mahajan, David Wetherall, and Tom Anderson. Understanding BGP misconfiguration. In *Proc. ACM SIGCOMM*, pages 3–17, Pittsburgh, PA, August 2002.

[31] Zhuoqing Morley Mao, Ramesh Govindan, George Varghese, and Randy Katz. Route Flap Damping Exacerbates Internet Routing Convergence. In *Proc. ACM SIGCOMM*, Pittsburgh, PA, August 2002.

[32] David Moore, Geoffrey Voelker, and Stefan Savage. Inferring Internet denial of service activity. In *Proc. 10th USENIX Security Symposium*, Washington, DC, August 2001.

[33] Vern Paxson. Strategies for sound Internet measurement. In *Proc. Internet Measurement Conference*, Taormina, Italy, October 2004.

[34] B. Quoitin and O. Bonaventure. A cooperative approach to interdomain traffic engineering. In *1st Conference on Next Generation Internet Networks Traffic Engineering*, Rome, Italy, April 2005.

[35] B. Quoitin, L. Iannone, C. de Launois, and O. Bonaventure. Evaluating the benefits of the locator/identifier separation. In *Proceedings of MobiArch 2007 (ACM SIGCOMM Workshop)*, Kyoto, Japan, August 2007.

[36] RouteScience. Whitepaper available from `http://www.routescience.com/technology/tec_whitepaper.html`.

[37] Justin Ryburn. Re: possible l3 issues, February 2004. `http://www.merit.edu/mail.archives/nanog/2004-02/msg00814.html`, plus private communication from list members who wished to remain anonymous.

[38] Stefan Savage, Tom Anderson, et al. Detour: A Case for Informed Internet Routing and Transport. *"IEEE Micro"*, 19(1):50–59, January 1999.

[39] Neil Spring, Ratul Mahajan, and Tom Anderson. Quantifying the causes of path inflation. In *Proc. ACM SIGCOMM*, Karlsruhe, Germany, August 2003.

[40] Neil T. Spring, David Wetherall, and Tom Anderson. Scriptroute: A public internet measurement facility. In *Proc. 4th USENIX Symposium on Internet Technologies and Systems (USITS)*, Seattle, Washington, March 2003.

[41] Shu Tao, Kuai Xu, Ying Xu, Teng Fei, Lixin Gao, Roch Guerin, Jim Kurose, Don Towsley, and Zhi-Li Zhang. Exploring the performance benefits of end-to-end path switching. In *IEEE International Conference on Network Protocols (ICNP)*, Berlin, Germany, October 2004.

[42] R. Teixeira, K. Marzullo, S. Savage, and G.M. Voelker. In Search of Path Diversity in ISP Networks. In *Proc. ACM SIGCOMM Internet Measurement Conference*, Miami, FL, October 2003.

[43] Vijay Vasudevan, David G. Andersen, and Hui Zhang. Understanding the AS-level path disjointness provided by multi-homing. Technical Report CMU-CS-TR-141, Carnegie Mellon University, July 2007.

[44] Wen Xu and Jennifer Rexford. MIRO: Multi-path Interdomain ROuting. In *Proc. ACM SIGCOMM*, Pisa, Italy, August 2006.

[45] Xiaowei Yang. NIRA: A New Internet Routing Architecture. In *ACM SIGCOMM Workshop on Future Directions in Network Architecture*, Karlsruhe, Germany, August 2003.

[46] Xiaowei Yang, David Wetherall, and Thomas Anderson. Source selectable path diversity via routing deflections. In *Proc. ACM SIGCOMM*, Pisa, Italy, August 2006.

[47] Xinyang Zhang, Paul Francis, Jia Wang, and Kaoru Yoshida. Scaling IP routing with the core router-integrated overlay. In *IEEE International Conference on Network Protocols (ICNP)*, Santa Barbara, CA, November 2006.

[48] D. Zhu, M. Gritter, and D. Cheriton. Feedback based routing. In *Proc. 1st ACM Workshop on Hot Topics in Networks (Hotnets-I)*, Princeton, NJ, October 2002.