Informed Detour Selection Using iPlane Helps Reliability

Boulat A. Bash

Technical Report UM-CS-08-26

Computer Science Department
140 Governors Drive
University of Massachusetts
Amherst MA 01003 USA

May 20, 2008

Abstract

In this report we propose to use the daily traceroute data generated by *iPlane* [9] to improve the reliability of the Internet paths. Previous work put forth a simple idea of using an *overlay* network of intermediary detour nodes that can be used to route around failures on direct Internet paths [1]. We provide the mechanism for choosing these intermediary nodes.

The underlying idea of our work is that knowledge of the point-of-presence (PoP) path between the source and destination as well as the PoP paths between potential detour nodes and destination can be used to pick the intermediary for routing around failures on the direct path. We leverage the existing iPlane infrastructure and perform an experiment using PlanetLab [3]. We obtain a substantial improvement in path availability over the state-of-the-art using the data from iPlane.

1 Introduction

As the Internet continues to evolve, reliability remains a key issue. Research has shown that the Internet in its current form fails to achieve the “five nines” (99.999%) of connection reliability that the public switched telephone network (PTSN) demonstrates [7], realizing between 96.7% and 98.5% reliability according to various studies [4, 10, 12]. There are a number of reasons for this, including server failures, router misconfigurations, and long BGP response times. One may say that this is to be expected of a network that provides a best-effort datagram service where most of the intelligence lies at the end-hosts (as opposed to the network itself, like in PTSN.) However, this does not stop the research community from attempting to mollify the situation.

There exists a substantial body of research on improving the reliability (and performance) of the Internet. Essentially, there are two main approaches for making the Internet service more reliable: adding server redundancy and path redundancy. Note that these methods are complimentary, and that if one adds redundant servers in topologically distinct regions of the Internet, one also increases path redundancy. Server redundancy is mainly achieved using content distribution networks (CDNs) [6]. While they are popular, unfortunately, there are only a few kinds of traffic that they can benefit, such as web page fetches.

Path redundancy can be applied to mitigate the Internet path failures. The idea is to navigate around the failed portion of the path using one (or more) *detour* nodes. Utilizing such path redundancy for reliability was first explored in a Resilient Overlay Network (RON) system [1] by Anderson *et al.* RON used aggressive monitoring of a relatively small overlay network (16 nodes) to recover from faults, and identify and exploit opportunities for performance improvement (such as utilizing violations of the triangle inequality to decrease latency.) Unfortunately, wide scale deployment of RON requires significant monitoring overhead, since the overlay continually probes the complete graph between all of its nodes. Thus, RON does not scale very well.

A light-weight detouring method that does not require path monitoring was proposed by Gummadi *et al.* and is called Scalable One-hop Source Routing (SOSR) [5]. Their main contribution is the demonstration that it is usually sufficient for failure recovery to attempt to route indirectly using 4 *randomly chosen* nodes from a sufficiently geographically distributed set of candidate detour nodes (they used 67 PlanetLab [3] machines as potential detour node set.) If one of the detour paths succeeds, then fault is avoided. Essentially, they utilize the idea of having multiple random choices used in load-balancing [2]. They present a Linux implementation and show that they can route around 56% of network failures.

We propose to combine the lightweight SOSR approach with the idea of *informed* (as opposed to random) choices of the detour nodes as done in RON. However, unlike RON, our system would not aggressively monitor the links in the Internet. Instead, we would like to exploit an existing and unrelated system for the data required to make an informed decision on which detour nodes to try. The system we will use is a PlanetLab-based Internet map project called iPlane [9]. iPlane provides two valuable services: IP prefix to Point-of-Presence (PoP) ID mapping and daily traceroute data from most PlanetLab sites to destinations in almost every PoP. Like in SOSR, we will use PlanetLab machines as our detour nodes. However, our client would rank the detour nodes by how much the PoP path (i.e. path according to the most current iPlane traceroute with IP addresses mapped to PoP IDs) from the detour node to destination overlaps with the PoP path from the client to the destination (generated either using client traceroute or iPlane and mapped to PoP IDs.) Less overlap is better. In this work we test two metrics for path overlap: count of common PoP IDs on two paths, and count of common PoP links on two paths— both yield almost identical results (as we will show in Section 3.) We will select the top k nodes as detour intermediaries, where $k = 1, 2, 3, 4, 5$ in this work, but can potentially be larger.

In the next section we will describe the methodology behind the experimental validation of our proposal, and in Section 3 we will demonstrate that our proposal yields significantly greater reliability benefit then SOSR. We will conclude with the discussion of further work in Section 4.

2 Methodology and the Experimental Design

In order to compare the performance of our informed detour selection to random selection, we implemented a distributed Internet measurement system on PlanetLab [3]. This section will describe the methods we used and our system design.

As of March 2008, PlanetLab consisted of more then 800 nodes worldwide. However, we found that most of those nodes were unusable for the purpose of our experiments, due to being unstable, offline, or having severe bandwidth limitations. In the database of PlanetLab nodes (`http://www.planet-lab.org/xml/sites.xml`) we found 267 nodes that met our criteria: $\geq$ 5 MB/s connection to the Internet, "production" status (as opposed to "alpha" and "beta"), and published RSA public key (we found that one can not log in to machines with missing RSA public key). Out of those 267, we were able to use 121 nodes as vantage points and intermediaries in our experiment (the rest were either offline, unreachable, did not accept our PlanetLab account as valid, or did not allow us permissions to run our programs.)

Each PlanetLab vantage point probed a subset of destinations randomly selected from a set of routers (.1 IP addresses) that are on the iPlane destination list. We restricted ourselves to routers and did not include end-hosts because we are mainly interested in the availability of *paths*. Routers are less likely to fail then end-hosts, and are more likely to consistently return probes. The destination lists were disjoint across our vantage points, thus, during normal operation, each destination was probed by one PlanetLab node.

The probing was done by pinging each destination every 15 seconds. The path was considered failed if two consecutive pings were missed. When the path failed, the vantage point responsible for probing the destination requested that PlanetLab nodes on its *intermediary set* ping the destination of the failed path and answer if the ping is successful. These requests were sent out every 15 seconds while the path to the destination was down. Intermediary set for each vantage point was the set of all PlanetLab nodes we were using excluding the vantage point and nodes on its site (for example, UMass site contains two PlanetLab nodes—neither would be included in either intermediary set) and containing only one randomly selected node per site (thus, none of our PlanetLab vantage points used both UMass nodes as intermediaries). These exclusions were made in order to minimize the load on PlanetLab nodes, as well as reduce the traffic generated by the experiment. Essentially, the set of intermediaries was the set of potential detour nodes that a client in a vantage point could use in case of a path failure. The programs deployed on PlanetLab were $5,643$ lines of Ruby code that ran using the `scriptroute` [11] tool.

While the experiment was running, we downloaded the daily traceroute files from the iPlane website. Each file was about 2.3 GB zipped and about 5 GB unzipped, in binary format. The total size of raw iPlane traceroute dataset was about 30 GB. We also downloaded the daily IP prefix to point-of-presence ID mappings. We preprocessed each iPlane traceroute file to select only the destinations in the set we used, as well as map each IP address in each relevant traceroute to its point-of-presence ID. Preprocessing reduced the total size of the iPlane traceroute dataset to about 2.8 GB.

We used two metrics to pick intermediaries for detouring: count of common PoPs on the paths from original vantage point and intermediary to the destination, and count of common PoP links on the paths from original vantage point and intermediary to the destination. When determining links, we removed the unknown hops returned by traceroute (i.e. "0.0.0.0" hops) as well as those hops that were not present in the iPlane IP-to-PoP mapping and used hops with known PoP IDs as link ends. Thus, if iPlane traceroute contained the following path: "IP in PoP-1, unknown IP, unknown IP, IP in PoP-2, IP in PoP-3", our system used {PoP-1, PoP-2} and {PoP-2,PoP-3} as links. We also did not use unknown hops in count of common PoPs. However, we did record unknown hops in the path hop count, which was used as the tie-breaker in our metric (we preferred shorter paths). We present our results next.

3 Experimental Results

We ran our experiments for 379 hours from 6:00 AM EST March 25th 2008 until 1:00 AM EST April 10th 2008. We found that not all of our PlanetLab vantage points would continuously map to a point-of-presence (PoP) of an iPlane vantage point. Thus, we were only able to use data from 46 vantage points. Each vantage point had access to between 83 and 97 intermediaries, since some PlanetLab nodes were reset or rebooted.

Our system recorded $54,793$ path outage events, with mean and median outage durations of $1,309.41$ and 120 seconds, respectively. Aggregate monitoring time across all paths was $4,858,248,555$ seconds (or about 154 years), accounting for PlanetLab node failures. Thus, aggregate path availability was 98.523% (i.e. on aggregate, the path was in outage 1.477% of the time.) The duration of outages that we witnessed had a heavy-tailed distribution, as evidenced by the scatter plot of the empirical complimentary cumulative distribution of the observed outage durations on Figure 1. This is consistent with the literature [4, 8]. There were some paths that were down for a long time: the longest outage we detected lasted 508,905 seconds, or 6 days 17 hours 21 minutes and 45 seconds.[1] We summarize the statistical properties of the outage time on Table 1.

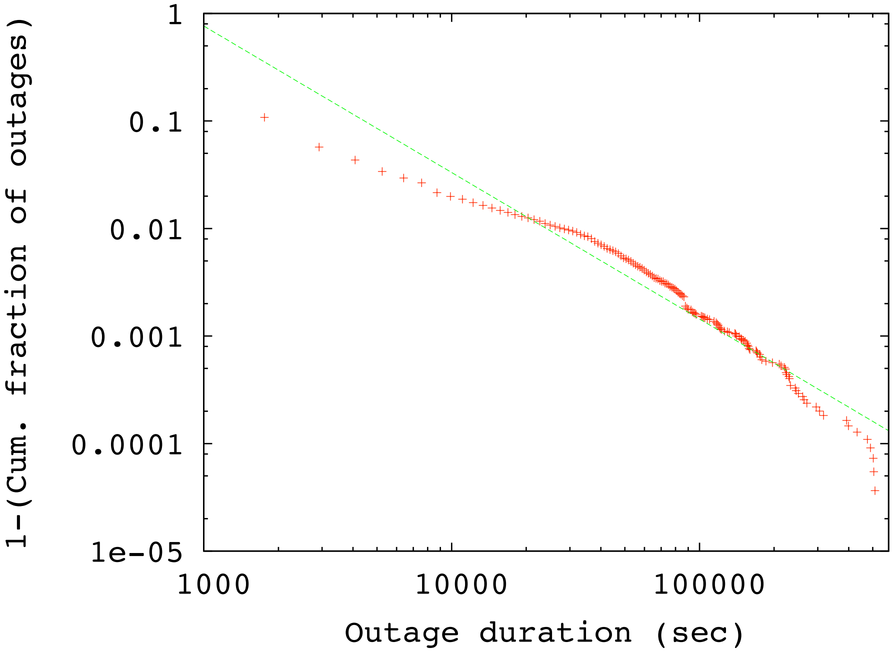

Figure 1: Empirical complimentary cumulative distribution of outage duration on $\log_{10}$-$\log_{10}$ scale, showing that it is heavy-tailed.

Table 1: Outage duration statistics

Number of events:	54,793
Mean:	1,309.41 sec
Median:	120 sec
Maximum:	508,905 sec
Standard deviation:	10,419.35
Skewness:	25.29

Our experimental framework records the intermediaries that were successful in reaching the destination. Thus, we could mimic a system where intermediaries are detour nodes used during path failures. Alternate path to a destination through one of the intermediaries was available during 44,276 out of 54,793 outages (80.8%). The mean number of intermediaries with an available path was 9.1, median number was 8. We can see from the histogram on Figure 2 that the distribution of the number of intermediate paths looks somewhat bimodal when paths exist

[1] One must note that there were 26 intermediaries that we could have used to route around the failure, including one whose path had the least number of common links with the direct path according to the iPlane traceroute data.

(i.e. for non-zero values of number of intermediaries with alternate paths), with peaks around 2 and 16. We do not know the reason for this. The paths through an intermediary whose path had least number of common PoPs and links with direct path were available $12,646$ times and $12,642$ times (both $\approx 23.1\%$), respectively. Table 2 summarizes the availability of paths through intermediaries.

Table 2: Availability of alternate paths

Number of outages:	$54,793$
Number of outages with alternate path:	$44,276$
Mean number of intermediaries with alternate path:	9.1
Median number of intermediaries with alternate path:	8

Number of best intermediaries	By common PoP count	By common link count
1	12,646 (23.1%)	12,642 (23.1%)
2	17,357 (31.7%)	17,231 (31.4%)
3	21,322 (38.9%)	21,449 (39.1%)
4	24,550 (44.8%)	24,568 (44.8%)
5	24,550 (44.8%)	24,568 (44.8%)

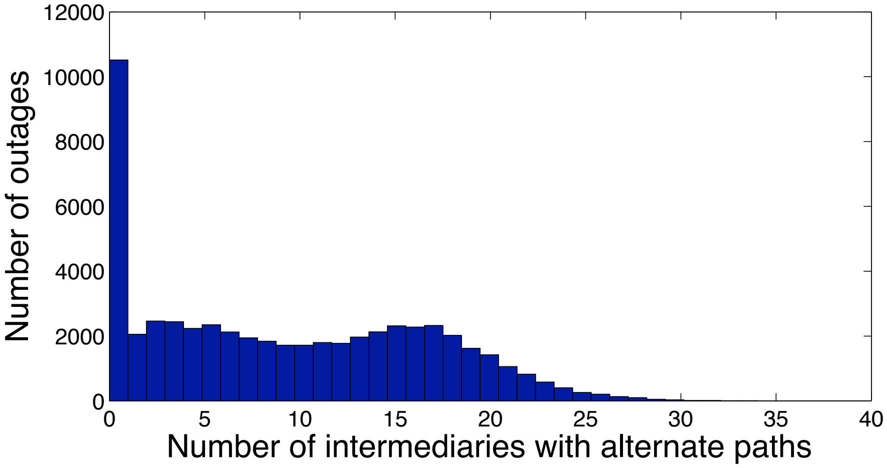

Figure 2: Histogram of path availability across outages

Now let us examine the performance of the SOSR-like system described in the introduction. Recall that the original SOSR system attempts to use 4 randomly-chosen detour nodes to recover from faults [5]. We select from one to five intermediaries in case of an outage and attempt to continue connection. Only if none of the intermediaries can reach the destination, the connection remains in outage. Otherwise, the system succeeds in preventing the break in the connection. The probability that the random selection of k intermediaries fails in the case of n potential intermediaries and $m \leq n$ intermediaries that have a working detour path to the destination can be expressed as follows:

$$\begin{aligned}\mathbf{P}\,(\text{failure using } k) &= \frac{\binom{n-m}{k}}{\binom{n}{k}} = && (1)\\ &= \prod_{i=0}^{k-1} \frac{n-m-i}{n-i} && (2)\end{aligned}$$

We use the equation (2) in our programs to compute the outage probability using SOSR-like random-k method. Table 3 illustrates that our informed selection methods substantially outperform the random intermediary selection. Note that the path availability for each method can be obtained by taking the complement of the corresponding outage probability given on Table 3. We are able to achieve "two nines" of reliability with either method and selecting just one intermediary node, while the random selection fell short even utilizing the power of five choices. We also note that the two informed methods are not substantially different in performance.

Table 3: Impact of intermediary selection methods on path outage

	Outage probability by selection method		
Num. intermediaries used	Random	Common PoP count	Common link count
0	1.477%	1.477%	1.477%
1	1.295%	0.757%	0.743%
2	1.147%	0.648%	0.654%
3	1.024%	0.566%	0.571%
4	1.023%	0.516%	0.527%
5	1.022%	0.516%	0.526%

Our system monitored a total of 4,269 paths. 1,715 (40.2%) of those paths did not experience a failure that we detected, and 328 (7.7%) paths had failures but there was no intermediary with available path to the destination (we suspect that these were due to destination failures as opposed to path failures.) Examination of the plot of the cumulative fraction of paths vs. their availability on Figure 3 reveals that informed method achieves considerable availability gains for the paths that are unavailable for substantial periods of time. In many cases of long-duration failures, informed method was able to find a working alternate path where random method would have had a high probability of failure due to the comparatively small fraction of intermediaries having good paths to those destinations.

4 Conclusion

We presented a proposal for improving the SOSR [5] system by replacing the existing random method with an informed mechanism for selection of detour nodes and showed that it leads to a substantial improvement in reliability. While we used iPlane [9] as the source of data driving the detour routing decisions, we note that our system is not tied to this specific service. We believe any source of coarse-grained path information could be used for the task (though, of course, one may get a different result.) We would be curious to see whether our results would improve with fresher route data, as right now we implicitly assume that PoP paths do not change frequently. We realize that we were somewhat forced into this assumption due to limitation of

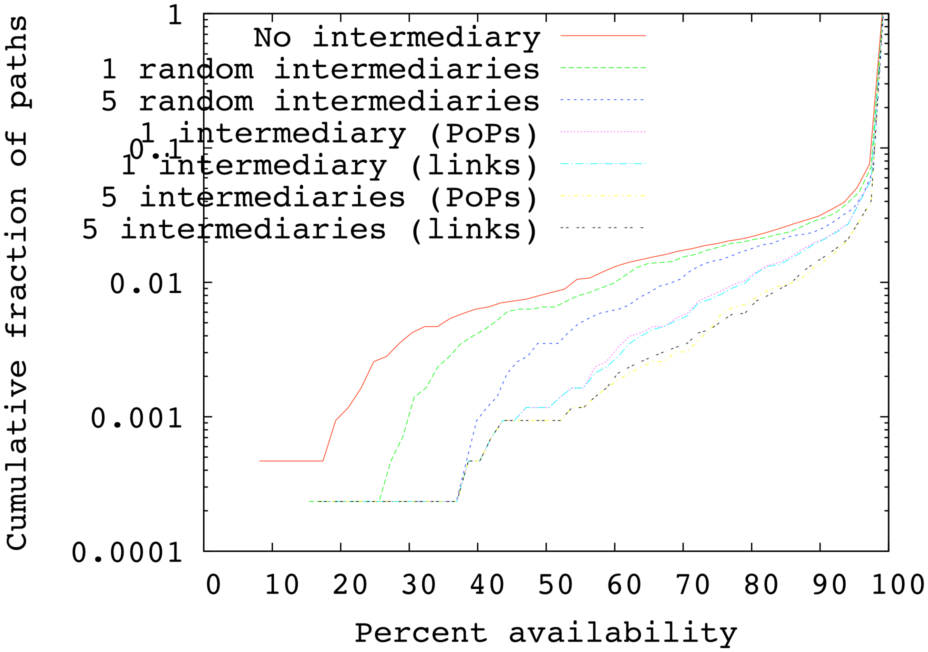

Figure 3: Cumulative fraction of paths vs. percent availability. Note that the curves corresponding to the informed methods are substantially to the right of the curves corresponding to random selection. This illustrates that the availability gain for the informed selection is on average greater in magnitude then of the gain for random selection.

iPlane architecture, and thus it would interesting to see the impact on our measurements if iPlane generated traceroutes twice or thrice a day instead of once as it currently does.

We would also like to implement an actual system that would query iPlane for the route data and test whether SOSR with informed detour selection will actually outperform SOSR with random detour selection.

5 Acknowledgement

We would like to thank Devesh Agrawal for allowing us to use his scriptroute and ruby-based software for probing destinations and his assistance in tuning it for our specific needs. Without him, this project would not have been possible. Our heartfelt gratitude also goes to Tyler Trafford who assisted us with all sorts of systems questions during all hours of the day and night. Also, we would like to thank Harsha Madhyastha for answering questions about iPlane.

References

[1] David Andersen, Hari Balakrishnan, Frans Kaashoek, and Robert Morris. Resilient overlay networks. *SIGOPS Oper. Syst. Rev.*, 35(5):131–145, 2001.

[2] J. Byers, J. Considine, and M. Mitzenmacher. Geometric generalizations of the power of two choices. In *Proc. of the 16th ACM Symp. on Parallel Algorithms and Architectures*, June 2004.

[3] Brent Chun, David Culler, Timothy Roscoe, Andy Bavier, Larry Peterson, Mike Wawrzoniak, and Mic Bowman. PlanetLab: an overlay testbed for broad-coverage services. *SIGCOMM Comput. Commun. Rev.*, 33(3):3–12, 2003.

[4] Michael Dahlin, Bharat Baddepudi V. Chandra, Lei Gao, and Amol Nayate. End-to-end WAN service availability. *IEEE/ACM Trans. Netw.*, 11(2):300–313, 2003.

[5] 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 *Proceedings of USENIX OSDI*, December 2004.

[6] Balachander Krishnamurthy, Craig Wills, and Yin Zhang. On the use and performance of content distribution networks. In *IMW '01: Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*, pages 169–182, New York, NY, USA, 2001. ACM.

[7] D. R. Kuhn. Sources of failure in the public switched telephone network. *Computer*, 30(4):31–36, April 1997.

[8] C. Labovitz, G. R. Malan, and F. Jahanian. Internet routing instability. *IEEE/ACM Transactions on Networking*, 6(5):515–528, October 1998.

[9] Harsha Madhyastha, Tomas Isdal, Michael Piatek, Colin Dixon, Thomas Anderson, Arvind Krishnamurthy, and Arun Venkataramani. iPlane: an information plane for distributed services. In *USENIX'06: Proceedings of the 7th conference on USENIX Symposium on Operating Systems Design and Implementation*, pages 26–26, Berkeley, CA, USA, 2006. USENIX Association.

[10] Vern Paxson. *Measurements and Analysis of End-to-End Internet Dynamics.* PhD thesis, U.C. Berkeley, 1997.

[11] Neil Spring, David Wetherall, and Tom Anderson. Scriptroute: A public internet measurement facility. In *Proc. of 4th USENIX Symposium on Internet Technologies and Systems*, pages 225–238, March 2003.

[12] Yin Zhang, Vern Paxson, and Scott Shenker. The stationarity of internet path properties: Routing, loss, and throughput. Technical report, ACIRI, May 2000.