Improving Delay Estimation with Path Stitching*

DK Lee, Keon Jang, Changhyun Lee, Sue Moon, and Gianluca Iannaccone†
Computer Science Department, KAIST
†Intel Research, Berkeley

Introduction

Massively distributed applications have become widespread in today's Internet. Popular examples are: peer-to-peer file sharing systems, CDNs (Content Distribution Networks), and massively multi-player online games. The number of such massively distributed applications is ever-growing, and their need for network-internal performance characteristics is ever-expanding. These applications would benefit greatly, if they could choose peers to communicate with based on latency, loss, bandwidth, topological proximity, or combinations of these metrics. A number of measurement infrastructures have been proposed and built to provide network-internal performance characteristics to a diverse set of distributed applications.

We envision a DNS-like system where applications send real-time performance measurement queries and receive information on desired metrics, without doing measurements by themselves. In this work, we propose a measurement retrieval infrastructure that unifies the existing measurement data and offers estimates for a variety of network performance metrics. We propose an alternative methodology to active measurements that utilizes existing heterogeneous measurements and extrapolates estimates for uncharted points in the Internet, and our work is orthogonal to existing body of work on network performance measurements. Our interests lie not in developing new topology or performance measurement techniques but in retrieving relevant data from existing measurements.

As a first step towards building such a measurement retrieval system, we investigate the feasibility of merging different types of measurements to improve upon the estimation accuracy without making extra active measurements. The main idea is to segment existing path measurements by the AS and incorporate AS-level routing information. Fittingly, we call our approach *path stitching*. It estimates the path and end-to-end latency between two arbitrary points in the Internet without incurring additional active measurements to either of the end host.

As a proof of concept, we compare estimated results with the real measurements. In this work, we demonstrate that despite many sources of errors in our approach, the estimates are very close to real live measurements. We conclude with a discussion on improving the accuracy and coverage of our approach.

*DK Lee, Keon Jang, Changhyun Lee, and Sue Moon were supported by the IT R&D program of MKE/IITA [2007-F-038-02, Fundamental Technologies for the Future Internet].

Related Works

Internet-Wide Measurements: Ark (the successor of Skitter) is a repository of traceroute outputs from tens of sources to thousands of destinations [11, 10]. A traceroute output from a source to a destination has hop-by-hop latency and path information. Missing information indicates loss. Its data collection spans 10 years and has been shared widely in the research community [1]. DIMES [20] expands the coverage of Skitter-like centralized data collection by endorsing help from volunteers who contribute traceroute outputs originating from their machines. As Barford *et al.* have analyzed, additional end-to-end path measurements are likely to have a good part of its path already in the measurement data set, bringing in only marginal utility in terms of network coverage [4]. In order to avoid measuring every path, NetQuest suggests a Bayesian experimental design in choosing active measurements for maximum information [21]. Donnet *et al.* proposes a tree-based exploration of the topology in order to reduce measurement traffic [7].

RouteViews [2] and RIPE RIS [19] are the major sources of Internet routing information. They store snapshots and updates of the BGP (Border Gateway Protocol) routing protocol contributed by many ISPs.

All these studies employ various underlying measurement techniques to derive Internet-wide performance characteristics. In our work, we focus on retrieving relevant data from these existing measurements.

Network Performance Estimation: Instead of taking measurements over every path, researchers have looked into estimation methodologies from a limited number of measurements [16, 17, 9, 23]. The main idea of their network embedding approaches lie in reducing the dimension of collected measurements to an Euclidean space of the Internet hosts; thus the name, network coordinates. PIC [5], NPS [15], and Vivaldi [6] take one step further. Instead of relying on an infrastructure of reference points, they use a decentralized approach and each participating

host measures and shares the information with other hosts (or peers). These measurement infrastructures, however, are usually stand-alone, use different performance metrics, employ various underlying measurement mechanisms, and often operate off-line only. Though diverse in underlying mechanisms, these systems have the common goal of providing network-internal characteristics to applications, and their measurements overlap significantly.

Network Performance Estimation As a Service: Internet end users or those massively distributed applications could benefit from the measurement studies, but data sharing with end users is yet limited. Aggarwal *et al.* have proposed an oracle service hosted by ISPs [3]. This oracle service ranks the queried peers according to certain metrics such as the number of AS hops to the users. ISPs do not need to measure performance, as they already have direct access to customers' bandwidth and link delay information. This service is mostly for peer-to-peer applications and relies on peer-to-peer applications to respect the ranking. Madhyastha *et al.* use systematic active measurements, construct an annotated map of the Internet, and return estimated latency, loss rate, and capacity in their iPlane system [12]. Our work is inspired by the above two services, but takes an extreme approach of taking no active measurements and relying on already available measurements or measurement infrastructures.

Delay Estimation through Path Stitching

In this work, we produce estimates for the path and latency between any two hosts in the Internet. Given the billions of hosts in today's Internet, any set of measurements can cover only a small portion of host-to-host pairs in the Internet. Our estimates would represent the uncharted points of the Internet. We use the traceroute outputs from Ark and combine them with BGP information from RouteViews and RIPE RIS. These are the largest data archives that hold constantly updated information about the IP and AS-level topology, and thus provide a good starting point for us to investigate the feasibility of our methodology.

We begin this section with an introduction to notations. Let $\mathcal{N}$ be a set of Internet hosts. If $n \in \mathcal{N}$, then n is an Internet host with the *intf(n)* number of network interfaces, and their IP addresses are n_1, n_2, ..., and $n_{intf(n)}$. It could be an end host or a router. We simply use n to denote an arbitrary interface address of n.

AS(n) :	origin ASes of a host *n*
path(n, m) :	a set of IP paths from host *n* to host *m*
ASpath(n, m) :	a set of AS paths from *AS(n)* to *AS(m)*

Due to MOASes (Multiple Origin ASes) often observed in routing tables, *AS(n)* could be more than one AS. The other two terms, *path(n, m)* and *ASpath(n, m)*, could also represent more than one path. A path in *path(n, m)* is a concatenation of host IP addresses, and a path in *ASpath(n, m)*, that of ASes.

$:A:$	a set of *path*(a_i, a_e), where a_i is an ingress router and a_e is an egress router of *A*.
$A::B$	a set of *path*(a_e, b_i), where a_e is an egress router of *A* and b_i is an ingress router of *B*.

The above two notations represent intra-domain paths and inter-domain paths, respectively. The first notation $:A:$ stands for a set of IP paths that connect AS A's ingress point (or ingress router) to its egress point. This is a set of paths that interconnect ingress and egress routers in the same AS. Every packet that reaches an ingress router of A and transits through it would choose one of the paths in $:A:$ to head out toward its destination. The second notation $A::B$ represents inter-domain paths that cross the boundary between the two ASes. For example, $A::B$ is a set of IP paths from an egress point of A to an ingress point of B.

We can stitch intra-domain and inter-domain paths to represent a longer path. For example, a concatenation of $:A:$ and $A::B$, denoted by $:A::B$, is a set of IP paths that traverse from an ingress point of A to an egress point of B. To find all IP paths that correspond to an AS path, $A_1A_2 \ldots A_n$, we stitch $:A_1:$, $A_1::A_2$, ..., $A_{n-1}::A_n$, and $:A_n:$, and obtain $:A_1::A_2: \ldots :A_{n-1}::A_n:$.

Index Building

The latency measurement data we use in this work consists of a large number of point-to-point traceroute outputs. An output from a single execution of the traceroute command has hop-by-hop IP addresses and delay measurements. In order to make the huge set of traceroute measurements searchable (*e.g.,* to find a matching AS path $ABCD$ in traceroute outputs), we first map all IP addresses in the outputs to AS numbers and build indices for search by the AS number. Here we face a scalability issue. If a traceroute path maps to an AS path $ABCX$, we could build indices for all possible partial paths, namely, $ABCX$, ABC, BCX, AB, BC, CX, A, B, C, and X. However, these indices require space in the order of $O(l^2)$, where l is the length of the AS path. Instead of building $O(l^2)$ indices, we choose to build only $O(l)$ indices of AB, BC, CX, A, B, C, and X. Whether we build $O(l^2)$ of indices or not, we are faced with the challenge of dealing with uncertainty in estimating inter-domain latency. Finding the longest matching partial AS path (in our example, ABC) in the database is likely to reduce error in latency estimation, but the overhead of $O(l^2)$ indices has led us to choose the latter approach.

We illustrate the index building process using two

traceroute outputs from host a to host b and host a' to host b'. The traceroute output of a to b returns: $a\ a_1\ a_2\ b_1\ b_2\ b$ with delay information at every hop. That of a' to b' returns: $a'\ a_1\ a_3\ b_3\ b_2\ b'$. Then the following information is stored:

$$
\begin{array}{rl}
:A: & (a\ ,\ a_2)\ \text{via}\ a_1\ \text{has delay}\ d_A \\
 & (a',\ a_3)\ \text{via}\ a_1\ \text{has delay}\ d'_A \\
A::B & (a_2,\ b_1)\ \text{has delay}\ d_{AB} \\
 & (a_3,\ b_3)\ \text{has delay}\ d'_{AB} \\
::B:: & (b_3,\ b')\ \text{via}\ b_2\ \text{has delay}\ d_B \\
 & (b_1,\ \ b)\ \text{via}\ b_2\ \text{has delay}\ d'_B
\end{array}
$$

Figure 1. Example Indices

Path Stitching

Given IP addresses of two hosts, we conduct the following three steps in order to estimate path and latency between them.

Step 1: Map the IP addresses to AS numbers
In order to map an IP address to an AS number, we use BGP routing tables. An individual IP address of a host n is mapped to the longest matching IP prefix in a BGP table, and we then take the last hop in the AS-PATH as the origin AS of this prefix.

Step 2: Infer AS paths between the ASes
We use Qiu and Gao's methodology to infer the AS paths between two ASes [18]. Their methodology exploits the AS paths appeared in BGP routing tables and infer AS paths based on theses known AS paths.

Step 3: Infer path and delay along the inferred AS paths
In this step, we utilize an existing set of path and delay measurement data. We have inferred an AS path between the two points of the Internet in Step 2, and we now try to find estimates for the inferred AS path. We first illustrate our methodology through an example.

We start with two hosts of interest, a_s and b_d. In Step 1, we map a_s to AS A and b_d to B. In Step 2, we infer that the AS-level path from A to B is simply AB. In Step 3, we dig through the indexed data find a latency estimate from a_s to b_d along the inferred AS path AB. Using the example in Figure 1, we come up with two stitched paths for $path(a_s, b_d)$:

$$
\begin{array}{ll}
(a\ , a_2), (a_2, b_1), (b_1,\ b) & \text{with delay}\ \ d_A + d_{AB} + d_B \\
(a', a_3), (a_3, b_3), (b_3, b') & \text{with delay}\ \ d'_A + d'_{AB} + d'_B
\end{array}
$$

We are given two paths and corresponding delay estimates to choose from. Which estimate to use is yet to be decided. In the next section, we present preliminary results on delay estimates and show our decision process.

Preliminary Results

For Steps 1 and 2, we use BGP routing table snapshots from RoueViews' BGP listener taken at 2 hour intervals. Routing table snapshots are stored in MRTD (Multithreaded Routing Toolkit Daemon) format. We also use snapshots from RIPE NCC Routing Information Service (RIS). RIS operates 14 monitoring points (rrc00 - rc07 and rrc10 - rrc15) and each monitoring BGP listener peers with different ASes.

In Step 3, we use publicly available measurement data from Ark, namely, cycle-20080407, cycle-20080408, and cycle-20080409 files. These three files combined contain one round of traceroute outputs from 22 sources to every /24 routable prefix. They contain more than 15 millions of traceroute outputs. As demonstrated through an example in the previous section, we partition the delay measurements of traceroute outputs and index them by the AS.

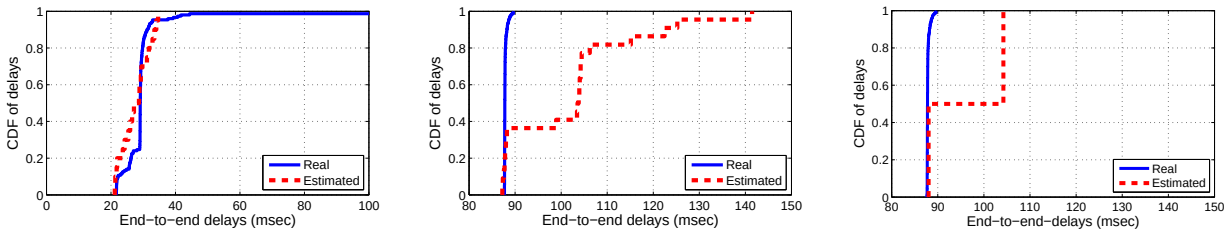
In order to compare estimated results with the real measurements, we have performed traceroute 50 times a day between 184 PlanetLab nodes for the three-day time period of Ark data. For every pair of PlanetLab nodes, we obtain a number of candidate paths stitched through our methodology. Some pairs have exactly one stitched path, while other pairs have tens or hundreds of paths.

In Figure 2, we compare the cumulative distribution of estimated delays with the real measurements for two pairs of hosts. The first pair in Figure 2(a) is between Planetlab nodes that are located in the same ASes as the Ark monitors, *cbg-uk* and *zrh-ch*. In this case, estimated delays align with real measurements very closely.

The other pair in Figure 2(b) is randomly selected from PlabnetLab nodes that are located not in the same AS as any Ark sources. The number of stitched paths is 22. In this figure, about 36% of estimated delays match the real measurements. In the distribution of the estimated delays, we observe two sharp jumps at around 88 and 105 msec. Apparently some stitched paths match the real path, and others not. As demonstrated in this example, not all stitched paths produce good estimates. In the next section we discuss the sources of error in our methodology and heuristics to address them.

Sources of Error

In each step, we face many sources of error. The main source of error – or limitation of our current solution – comes from the dynamic nature of routing in the Internet. Routing protocols detect changes and failures and constantly adjust packet forwarding paths in the network, as they are designed to. Using only past measurements, we are inherently constrained by the age of the collected data. For this work, we only keep the latest delay measurement in the indexed data from Ark. Our methodology should improve in estimation accuracy with more data sources,



(a) planetlab2.xeno.cl.cam.ac.uk → lsirextpc02.epfl.ch

(b) planetlab1.csail.mit.edu → planet2.scs.stanford.edu

(c) planetlab1.csail.mit.edu → planet2.scs.stanford.edu

Figure 2. Accuracy of estimations

and we leave the timeliness of our methodology for future work.

Other sources of error stem from limited measurements and approximation techniques. In Step 1, a prefix could be mapped to a set of ASes, if multiple ASes have announced as origin ASes for the prefix. Mao *et al.* have pointed out that this BGP table-based IP-to-AS mapping technique is not 100% accurate and most errors stem from route aggregation, interface numbering at AS boundaries, and routing anomalies [13, 14].

Inferring an AS-level path between two ASes without access to either of the AS is not simple. We use Qiu and Gao's methodology, KnownPath. It exploits the AS paths derived from BGP routing tables and infer AS paths by extending theses known AS paths [18]. Each inferred AS path is associated with two path attributes, unsure length and frequency index. Unsure length is the length of the the extended part, and is a measure of uncertainty in the inferred path. Frequency index indicates the likelihood of sure subpath to be used in the Internet. When KnownPath algorithm produces more than one $ASpath(x, y)$, we choose the shortest path with the minimum unsure length and higher frequency index.

In Step 3 of our methodology, we may have multiple stitched paths and need to select the closest path to the real path. Our first tie-breaking criteria is the destination prefix. The stitched paths all share the same AS path, but the BGP routing decision is actually made by the destination prefix. We choose only those stitched paths that lie in the same /24 prefix as the target destination or b_d in our example. When we apply this tie-breaking rule, we are left with only two stitched paths in Figure 2(c), a great reduction from 22 paths in Figure 2(b). However, one of the two values is still far from the real measurement.

Tie-breaking based on the destination prefix may return more than one path. In our second tie-breaking criteria, we check if each stitched segment is originally from a path towards the same /24 destination prefix. From the destination, we backtrack to the source of the stitched paths and see if each segment in the stitched path was originally from a traceroute output towards the same /24 prefix. We continue this backtracking to the source or until we are left with only one stitched path. However, this backtracking requires storage for all destination prefixes per path segment. We are in the process of implementing this technique.

So far, we have only discussed the cases when we find stitched paths from the indexed data. What if we do not have any stitched path? Between $17,879$ pairs of 184 PlanetLab nodes we conducted traceroute (we exclude pairs with unsuccessful traceroutes,) we have obtained stitched paths for only 4,091 pairs. For the remaining pairs, stitched paths do not exist for the following reasons:

- The source is not in the same AS with any measurement data. (Since Ark covers all routable /24 prefixes, the destination has at least one measurement data point or a path segment of the same AS in the indexed data.)
- There is no inter-domain or intra-domain path for some segment in the inferred AS path.

The challenge here is to stitch up path segments that do not match at the ending and beginning nodes. We are currently considering the following heuristics. Most IP addresses in traceroute outputs are interface addresses of routers. We identify IP addresses that belong to the same router [8] or the same Point-of-presence (PoP) [22]. This router and PoP-level clustering method is a well-grounded solution, but requires a massive amount of active probes to routers in consideration. Our alternative approach is to cluster two ending points based on their IP prefix proximity. We also allow reverse path segments for inter-domain paths. That is, if there is a path segment $A::B$, but not $B::A$ and the inferred AS path contains $B::A$, then we take the delay estimate from $A::B$. We plan to evaluate our heuristics for the $(17,829 - 4,091)$ pairs of hosts.

Concluding Remarks

In this work, we introduce our *path-stitching* method and outline challenges in dealing with sources of error. We

demonstrate that despite many sources of error in our approach, our estimates are close to real live measurements. We expect that, as a end-to-end path and latency prediction method, *path stitching* provides opportunities for users to query about other hosts that they have no direct access. We plan to incorporate data sets other than traceroute and BGP tables.

References

[1] Research and publications based on skitter data. `http://www.caida.org/tools/measurement/skitter/`.

[2] Advanced network technology center and University of Oregon. The RouteViews project. `http://www.routeviews.org`.

[3] V. Aggarwal, A. Feldmann, and C. Scheideler. Can ISPs and P2P users cooperate for improved performance? *SIGCOMM Computer Communication Review*, 37(3):31–40, 2007.

[4] P. Barford, A. Bestavros, J. Byers, and M. Crovella. On the marginal utility of network topology measurements. In *Proceedings of the IEEE INFOCOM*, San Francisco, USA, November 2001.

[5] M. Costa, M. Castro, A. Rowstron, and P. Key. PIC: Practical Internet coordinates for distance estimation. In *International Conference on Distributed Systems*, Tokyo, Japan, March 2004.

[6] F. Dabek, R. Cox, F. Kaashoek, and R. Morris. Vivaldi: A decentralized network coordinate system. In *Proceedings of the ACM SIGCOMM*, September 2004.

[7] B. Donnet, P. Raoult, T. Friedman, and M. Crovella. Deployment of an algorithm for large-scale topology discovery. *IEEE Journal on Selected Areas in Communications*, 24(12):2210–2220, 2006.

[8] R. Govindan and H. Tangmunarunkit. Heuristics for internet map discovery. In *Proceedings of the IEEE INFOCOM*, March 2000.

[9] Y. Hyun, A. Broido, and kc claffy. On third-party addresses in traceroute paths. In *Passive and Active Measurement Workshop 2003*, La Jolla, CA, April 2003.

[10] Y. Hyun, B. Huffaker, D. Andersen, E. Aben, C. Shannon, and M. Luckie. The CAIDA IPv4 Routed /24 Topology Dataset - April 2008. `http://www.caida.org/data/active/ipv4_routed_24_topology_dataset.xml`.

[11] kc claffy and Tracie E. Monk and Daniel McRobb. Internet tomography. *Nature, Web Matters*, January 1999.

[12] H. V. Madhyastha, T. Isdal, M. Piatek, C. Dixon, T. Anderson, A. Krishnamurthy, and A. Venkataramani. iPlane: An information plane for distributed services. In *Proceedings of the OSDI*, November 2006.

[13] Z. M. Mao, D. Johnson, J. Rexford, J. Wang, and R. Katz. Scalable and accurate identification of AS-level forwarding paths. In *Proceedings of the IEEE INFOCOM*, Hong Kong, China, March 2004.

[14] Z. M. Mao, J. Rexford, J. Wang, and R. Katz. Towards an accurate AS-level traceroute tool. In *Proceedings of the ACM SIGCOMM*, Karlsruhe, Germany, August 2003.

[15] T. E. Ng and H. Zhang. A network positioning system for the interent. In *Proceedings of USENIX Conference*, June 2004.

[16] T. S. E. Ng and H. Zhang. Predicting Internet network distance with coordinates-based approaches. In *Proceedings of the IEEE INFOCOM*, New York, USA, June 2002.

[17] M. Pias, J. Crowcroft, S. Wilbur, T. Harris, and S. Bhatti. Lighthouses for scalable distributed location. In *In International Workshop on Peer-To-Peer Systems,*, Berkeley, USA, February 2003.

[18] J. Qiu and L. Gao. As path inference by exploiting known as paths. In *Proceedings of the IEEE GLOBECOM*, San Francisco, USA, November 2006.

[19] RIPE NCC. The Routing Information Service. `http://www.ripe.net/ris`.

[20] Y. Shavitt and E. Shir. DIMES: Let the Internet measure itself. *SIGCOMM Computer Communication Review*, 35(5):71–74, 2005. `http://www.netdimes.org`.

[21] H. H. Song, L. Qiu, and Y. Zhang. Netquest: A flexible framework for large-sclae network measurement. In *Proceedings of the ACM SIGMETRICS*, 2006.

[22] N. Spring, R. Mahajan, and D. Wetherall. Measuring ISP topologies with Rocketfuel. In *IEEE/ACM Transactions on Networking*, 2004.

[23] B. Wong, A. Slivkins, and E. G. Sirer. Meridian: A light weight network location service without virtual coordinates. In *Proceedings of the ACM SIGCOMM*, August 2005.