Internet Sibilla: Utilizing DNS for Delay Estimation Service*

Keon Jang, DK Lee, Sue Moon
Computer Science Department, KAIST
{keonjang, dklee, sbmoon}@an.kaist.ac.kr

Gianluca Iannaccone
Intel Research, Berkeley
gianluca.iannaccone@intel.com

1. INTRODUCTION

Massively distributed applications are popular in today's Internet. To improve the end-user experience, they require constantly updated information about the network-internal performance characteristics, such as RTT, effective bandwidth, IP hop count, and loss rate. Knowledge of network-internal characteristics allow distributed applications to solve commonly encountered problems, such as nearest neighbor discovery, leader node selection, and optimal distribution tree organization. Today's Internet does not provide any such information, and applications and new services resort often perform their own measurement to obtain necessary information.

To provide network-internal characteristics, a number of measurement systems have been proposed and implemented [2, 3, 9–12, 14]. Despite their novel methodologies of Internet-wide measurement and network performance estimation, they are yet to be widely adopted. These systems are usually stand-alone, use different performance metrics, employ various underlying measurement mechanisms, and often operate off-line. Application developers have to import a library and modify the code, or even build the proposed system by themselves. An easy interface to a new system is the key in the system's success.

In this work, we propose to design an end-to-end delay estimation service, called Sibilla. The novelty of our system is that we utilize existing measurements and an infrastructure. Our delay estimation is based on a simple idea of path stitching [7]. Path stitching takes CAIDA's Ark traceroute repository [5] and BGP tables from RouteViews [1] and RIPE RIS [13], segments the path and delay information of traceroute by the AS, stitches up segments by the AS path, when a query arrives, and produces an estimated delay. To provide the path-stitching as a service, we design and deploy our estimation service system on top of the Domain Name System (DNS). We deploy a Sibilla DNS server that receives a query between two hosts and replies with an estimated delay through the DNS interface. We take advantage of the fact that local DNS cache servers are distributed globally and exist in almost all ASes and improve the accuracy of our path stitching algorithm.

*Keon Jang, DK 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].

ACM CoNEXT 2008, December 9-12, 2008, Madrid, SPAIN

2. DNS-BASED SYSTEM DESIGN

The basic idea of our service system is to formulate a delay query between two hosts as a DNS query and let our Sibilla DNS server answer it. For example, when a user wants to estimate delay between two points A and B, the user submits a DNS query for A_B.latency.internet-sibilla.com. The local DNS server forwards the query to our Sibilla DNS server, which in turn replies with delay estimation as described in Figure1.

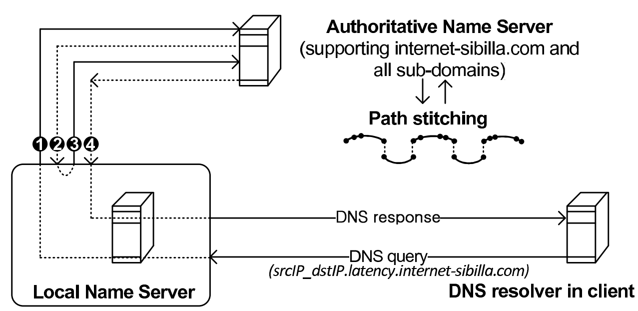

Figure 1: Overview of SIBILLA

We consider two different types of a query as in Table 1. The first type is a basic query about delay between two hosts. The DNS query includes two host addresses and the Sibilla DNS server replies with an estimated delay. The second type is used for nearest neighbor selection based on the latency. If a end host embeds more than two IP addresses in the query,

Type	Example
Latency query	srcIP_dstIP.latency.internet-sibilla.com
Server selection	srcIP_dstIP1_dstIP2.latency.internet-sibilla.com

Table 1: Query Types

the Sibilla DNS server processes the query as one source IP address and multiple destination IP addresses, estimates delay from the source to all the destination addresses, and returns an answer of the destination address with the smallest delay. DNS replies for both types of queries are type A.

Data Type	Total AS	Transit AS	Stub AS
Ark	14378	4418	9960
BGP	28244	4847	23397

Table 2: Number of ASes in Ark and BGP data

At the core of our system lies the path stitching algorithm. The input to the algorithm is traceroute outputs. The accuracy of the algorithm is inherently dependent on traceroute. Traceroute often fails to reach a destination [8]. We present the number of ASes in BGP tables from RouteViews and one daily Ark data set in Table 2. One daily Ark data set includes traceroutes from 18 monitors to every /24 prefixes appeared in the BGP routing tables. There we see that the Ark data set has only half the number of ASes as in the BGP tables. If we split the ASes into transit and stub*, 93% of transit ASes in the BGP tables appear in the Ark data set, and the majority of ASes not in the Ark data set are basically stub ASes. Further investigation reveals that more than 89% of the missing ASes ($12,382$ out of $13,866$) correspond to traceroutes that did not reach their intended destinations and returned incomplete results. That is, the key point to improve the accuracy in our path-stitching algorithm is to address this shortcoming.

In order to add more ASes in our delay database, we take advantage of the fact that DNS servers are globally distributed and preside in almost every AS. We entice local DNS servers to submit a request twice so that we obtain delay between the local DNS server and the Sibilla DNS server. We assume that two requests are processed at a local DNS server back-to-back and we can take the round-trip delay from the Sibilla DNS server.

Figure 1 illustrates this mechanism: (1) A local DNS server sends a DNS request to the Sibilla DNS server, (2) the Sibilla DNS server replies with an type NS (Name Server) Resource Record (RR), (3) the local DNS server queries again, and (4) the Sibilla DNS server returns an answer with an type A (Address) RR. Assuming the local DNS send the second query immediately after it receives the type NS RR, RTT is the time difference between the first reply and the second query at the Sibilla DNS server. The benefit of this method is that it works even when ping or traceroute to a client's network is blocked. Using latency to a local DNS server instead of directly to clients is well studied and shown to work [4,6]. We are in the process of evaluating the accuracy of this method.

*We call an AS transit if it appears in any AS path (not the first nor the last AS) in a BGP table; otherwise, stub.

3. SUMMARY AND FUTURE WORK

In this poster, we have outlined our DNS-based Sibilla system. The beauty of our system lies in the simplicity of the service system design. Any improvement of the path stitching algorithm in terms of added measurements and better estimation algorithms can be immediately reflected in a deployed system, as the DNS interface cleanly separates the user interface and the estimation mechanism.

Still, several challenges remain. To change our delay estimation algorithm from the current off-line version to serve thousands of queries a second, we plan to profile our algorithm and employ appropriate accelerating measures. As the accuracy of archived data decays over time, we need a quality control mechanism to ensure the freshness of data.

Our goals are to make our service system (i) efficient to replace the need for in-house development of a measurement infrastructure; (ii) extensible to support new applications that track different internal metrics of the Internet; and (iii) scalable to sustain a large and growing user base.

The extension of this work would apply to the Future Internet, where this type of service might be integrated into the core network design. We also hope our system works as a building block of future Internet applications to help developers with a wealth of information about the network performance and to make their applications more intelligent and productive.

4. REFERENCES

[1] http://www.routeviews.org.
[2] http://www.ripe.net/ris.
[3] M. Costa, M. Castro, A. Rowstron, and P. Key. PIC: Practical Internet coordinates for distance estimation. In *ICDCS*, Japan, March 2004.
[4] F. Dabek, R. Cox, F. Kaashoek, and R. Morris. Vivaldi: A decentralized network coordinate system. In *Proc. of the ACM SIGCOMM*, September 2004.
[5] K. P. Gummadi, S. Saroiu, and S. D. Gribble. King: estimating latency between arbitrary internet end hosts. *SIGCOMM Comput. Commun. Rev.*, 32(3):11–11, 2002.
[6] Y. Hyun, B. Huffaker, D. Andersen, E. Aben, C. Shannon, and M. Luckie. The CAIDA IPv4 Routed /24 Topology Dataset - April 2008.
[7] K. L. Johnson, J. F. Carr, M. S. Day, and M. F. Kaashoek. The measured performance of content distribution networks. *Computer Communications*, 24:202–206, 2001.
[8] D. Lee, K. Jang, C. Lee, S. Moon, and G. Iannaconne. Improving delay estimation with path stitching. In *Proc. of Conference Future Internet*, 2008.
[9] M. Luckie, Y. Hyun, and B. Huffaker. Traceroute probe method and forward ip path inference. In *Proc. of ACM SIGCOMM IMC*, Greece, Oct. 2008.
[10] H. V. Madhyastha, T. Isdal, M. Piatek, C. Dixon, T. Anderson, A. Krishnamurthy, and A. Venkataramani. iPlane: An information plane for distributed services. In *Proc. of the OSDI*, November 2006.
[11] T. E. Ng and H. Zhang. A network positioning system for the Internet. In *Proceedings of USENIX Conference*, June 2004.
[12] 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.
[13] M. Pias, J. Crowcroft, S. Wilbur, T. Harris, and S. Bhatti. Lighthouses for scalable distributed location. In *Proc. of IPTPS*, Berkeley, USA, February 2003.
[14] B. Wong, A. Slivkins, and E. G. Sirer. Meridian: A light weight network location service without virtual coordinates. In *Proc. of the ACM SIGCOMM*, August 2005.