Segmentation of Internet Paths for Capacity Estimation

Matthew Luckie, Tony McGregor

I. Introduction

The ability to identify the bottleneck capacity and the position in a path where a bottleneck occurs is helpful for network operators in understanding the performance seen on a path. Current capacity estimation techniques measure network paths with varying levels of accuracy, speed, and robustness. Tools like `pathrate` have become progressively more sophisticated and accurate at measuring the capacity of a tight link, but do not provide any indication of where the tight link is located in the path. `pathchar`-like techniques [1], [2] attempt to measure the capacity of each hop by exploiting the Time To Live (TTL) value, but it is difficult to isolate the delay contributed by a specific link when a noisy link preceeds the link measured.

Some current techniques and tools for measuring link capacities using techniques similar to `pathchar` are not accurate beyond a few hops, can place significant load on the links probed, and can take a significant length of time to run [2]. In a 1997 Mathematical Sciences Research Institute presentation [1, slide 12], Van Jacobson identified the need to be able to segment a path into hops in order to isolate the behaviour of a particular segment of interest as important for future research.

This paper describes a technique to segment a path into isolated links for capacity estimation purposes, by timestamping the probe packets at the ingress interface of each router in the path. The technique uses the IP Measurement Protocol (IPMP) [3] to obtain timestamps at each point in the network, but could use any other tracing protocol that provides a timestamp of sufficient resolution. The technique can be used to estimate the capacity of each hop by a method that isolates each hop in the path in an end-to-end measurement. This technique allows for forward and reverse path measurement and does not require the timestamp sources at each router to be synchronised.

II. The Method

Our method to estimate the capacity of each segment in a path uses a variation of the packet tailgating method first used in [2]. Rather than send a large packet with a TTL set to expire at a particular point in a path, we send a large packet that a smaller second packet has to queue behind as they pass through the network. The capacity of a particular segment can be estimated by the difference in time between the last bit of the first packet and the last bit of the second packet.

A packet P has timestamps inserted at each hop h in the network and can be thought of as being held in an array, so that the timestamp inserted into the packet P at hop h is $P_{[h]}$. Given a packet-pair that consists of P and P', where P is of size S and P' is of size S' where $S > S'$, and we send enough packet-pairs through the network such that at some point each segment will have at least one packet-pair traverse it back-to-back. Then the capacity C of each segment h in a path that consists of H hops can be calculated as follows:

$$C = \min_{h=0 \ldots H} \left\{ \frac{S - S'}{P_{[h]} - P'_{[h]}} \right\} \quad (1)$$

The main advantage of this method is that because the behaviour of each segment is confined to just that segment, so that the effect a noisy link has on subsequent links is minimised. The main limitation of this method is that we require the second packet in the pair to be received by a router in its entirety before the last bit of the first packet has been forwarded to the next hop, so that the packet-pair does not get separated except when caused by cross-traffic. Given an egress link n times faster than the ingress link to a router, then S' must be no larger than S/n. Unless the second packet regains its position behind the first packet, the packet-pair will not be able to capture the capacity of links after the separation. If the first packet has to queue behind other packets at a router after the separation, this may provide an opportunity for the second packet to regain its position behind the first packet. This limitation is not unique to the IPMP application of the packet-tailgating technique.

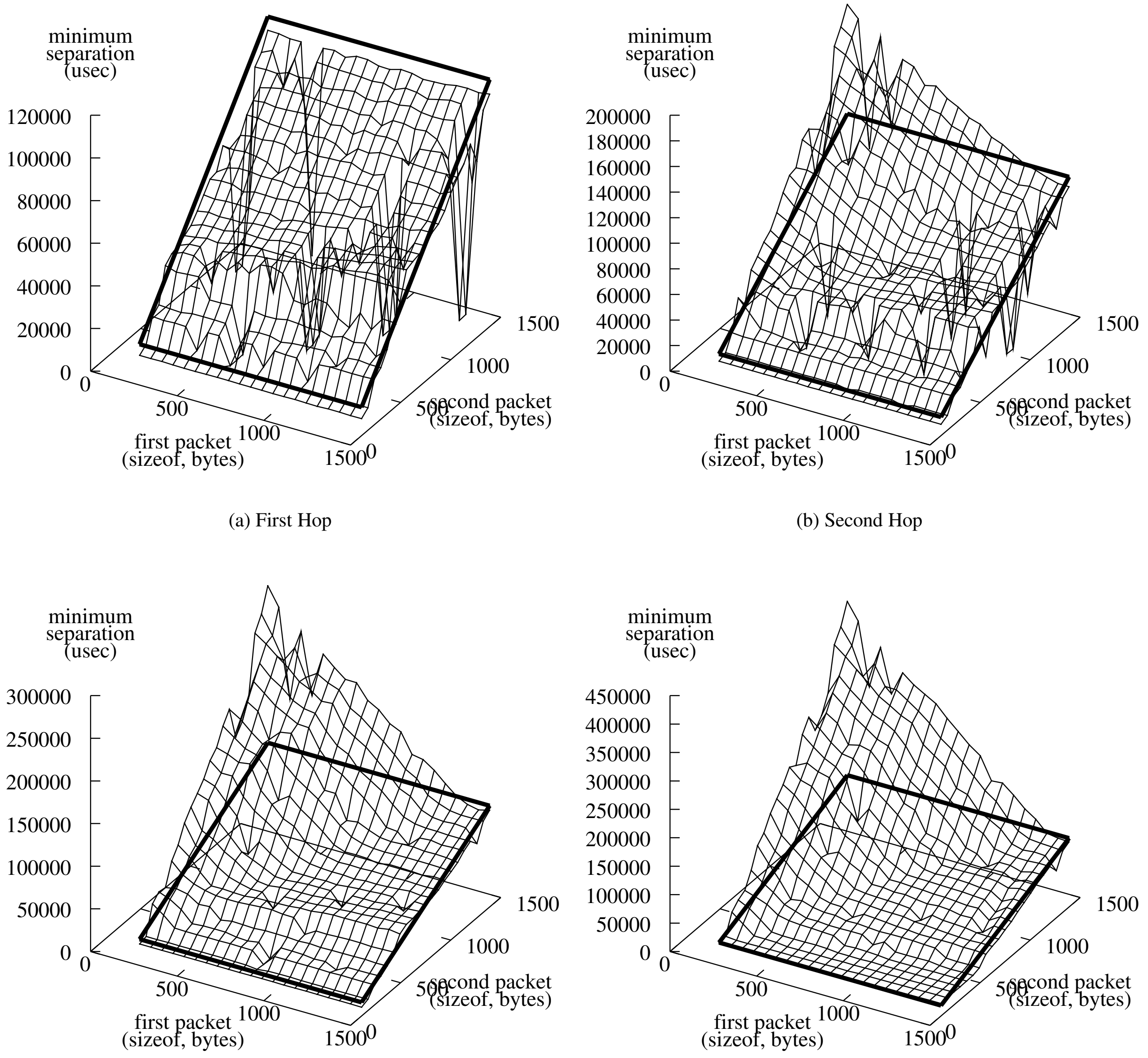
III. Results

We have implemented and tested this capacity estimation technique in a controlled environment (lab). As we need to modify the forwarding path to insert timestamps in specially marked packets which we cannot do with a traditional router, we have written kernel modifications to the Linux and FreeBSD kernels. We have completed some preliminary experiments of this technique with a tool we call `ipmp_pathchar`, which sends IPMP echo packets through the network. We have also developed a small network for experimentation purposes known as the Waikato Applied Network Dynamics (WAND) Emulation Network. This network consists of 24 computers networked together with crossover cables. Each of these computers runs Linux-2.4.20 with the PPSkit patch [4], a P2-350mHz processor, and an Intel Pro100-based 100mbit network interface card.

Figure 1 shows the minimum separation seen by sets of 200 IPMP packet-pairs of various combinations of first-packet size and second-packet size at each point in the network as they progress through the first 4 segments of an 8 segment round-trip path. In this experiment, the path consists entirely of 100mbit links and has no cross traffic. The thick black box at the base of each graph represents the time to serialise the second packet.

The authors are with the Waikato Applied Network Dynamics (WAND) group, University of Waikato, Hamilton, New Zealand and the National Laboratory for Applied Network Research Measurement and Network Analysis Group (NLANR/MNA). NLANR/MNA is supported by the National Science Foundation (NSF) under cooperative agreement no. ANI-0129677.

Email: mjl@luckie.org.nz, tonym@cs.waikato.ac.nz



(a) First Hop

(b) Second Hop

(c) Third Hop

(d) Fourth Hop

Fig. 1. Experiments with varying first and second packet size combinations, conducted on the WAND Emulation Network, using our `ipmp_pathchar` tool. The thick black box at the base of each graph represents the time to serialise the second packet. Note: the angle of the black box decreases as the Z-scale on the graphs is increased; the seralisation time does not change.

The graphs indicate that as long as the second packet is smaller than the first packet, then we can use the minimum separation of the second packet from the first to measure the capacity of that segment. The 'lift' in the graph in the Z axis in the area behind X=0, Y=1500 shows that the second packet becomes further away from the first packet as they progress through the network.

ACKNOWLEDGEMENTS

The authors are thankful for the editorial assistance provided by Maureen C. Curran.

REFERENCES

[1] V. Jacobson, "pathchar - a tool to infer characteristics of Internet paths," Apr. 1997, `ftp://ftp.ee.lbl.gov/pathchar/msri-talk.pdf`.

[2] K. Lai and M. Baker, "Measuring link bandwidths using a deterministic model of packet delay," in *Proceedings of SIGCOMM '00*, Stockholm, Sweden, Aug. 2000.

[3] M.J. Luckie, A.J. McGregor, and H-W. Braun, "Towards improving packet probing techniques," in *Proceedings of the ACM SIGCOMM Internet Measurement Workshop*, San Francisco, CA, Nov. 2001, pp. 145–151.

[4] U. Windl, "Implementation of nanosecond time and a PPS API for the Linux 2.4 kernel," `http://www.kernel.org/pub/daemons/PPS/`.