Spectroscopy of traceroute delays

Andre Broido, Young Hyun and kc claffy

Cooperative Association for Internet Data Analysis
SDSC, University of California, San Diego
{broido, youngh, kc}@caida.org

Abstract. [1] We analyze delays of traceroute probes, i.e. packets that elicit ICMP TimeExceeded messages, for a full range of probe sizes up to 9000 bytes as observed on unloaded high-end routers. Our ultimate motivation is to use traceroute RTTs for Internet mapping of router and PoP (ISP point-of-presence) level nodes, including potentially gleaning information on equipment models, link technologies, capacities, latencies, and spatial positions. To our knowledge it is the first study to examine in a reliable testbed setting the detailed statistics of ICMP response generation.

We find that two fundamental assumptions about ICMP may not hold in some cases in modern routers, namely that ICMP delays are a linear function of packet size and that ICMP generation rate is equal to the capacity of the interface on which probes are received. The primary causes of these violations appear to be internal segmentation of packets into cells and limiting of ICMP packet rates and bit rates inside a router. Our results suggest that the linear model of packet delay as a function of packet size merits revisiting for certain router models and time resolutions. Our findings also suggest possibilities of developing new techniques for bandwidth estimation and router fingerprinting.

1 Introduction

Remote network mapping is usually done via active measurement. Generally a measurement host sends packets that trigger ICMP replies from routers, and the reply information is integrated into a map. ICMP *time exceeded*, *echo reply* and *port unreachable* responses are commonly elicited for this purpose.

An ICMP reply carries binary ("host is alive"), discrete ("9 hops away") and temporal ("replied in 15 ms") data. The network delay, e.g. round trip time (RTT), is potentially the richest source of information about routers in the path. However, extracting the useful components from a delay value is difficult, since not only are the delay summands unavailable but even their statistics and their dependence on other factors are unknown.

In the common linear model, packet delay is split into three summands, with one being proportional to packet size. Specifically, the delay, d, is modeled as follows:

$$d = ax + b + \xi \qquad (*)$$

[1] This is an updated version of the paper published in *Proceedings of PAM 2005* (in the hardcopy version published by Springer; the electronic proceedings include this updated version) that correctly takes into account some specifics of timestamping, previously unknown to the authors, in the particular version of the firmware used in the Dag GE card of our experiments.

where a and b are positive real constants, x is the size of the packet or frame, and ξ is a positive random variable ("residual delay") that can be arbitrarily close to 0. This representation implies that $d = ax + b$ is a tight lower bound for all observed delays. Most network spectroscopy and bandwidth estimation experts assume that delay is a linear function of packet size, [1] [2] [3].

Our main goal in this study is to test the validity of this linear model, at least with respect to delays seen in ICMP responses (we do not cover forwarding delay in this study). Our underlying motivation is to find ways of using traceroute RTTs to:

- construct router and PoP-level Internet maps [4] [5]
- obtain metric maps with link latencies and capacities
- enable *user-level path diagnosis* [6]
- improve the integrity of variable-size bitrate estimation tools [7]; and
- fingerprint routers.

For example, one approach to identifying a PoP would be to look at traceroute paths that branch between backbone and access routers. Given that the routing to external destinations is common among all routers within a PoP, return paths to the monitor will be the same. One could thus use the topological closeness of forward paths together with the numeric closeness of RTTs to identify interfaces that belong to the same PoP. This aggregation technique requires precise knowledge of typical latencies across a PoP, as well as how often and for how long ICMP TimeExceeded generation can be delayed.

A typical traceroute covers 14–20 hops [8], and during a traceroute all but the last hop respond with an ICMP TimeExceeded packet. The last hop responds with an ICMP EchoReply or ICMP PortUnreachable. We will discuss properties of delays obtained from TimeExceeded packets in detail. We hope to report on destination-based (EchoReply, PortUnreachable) ICMP delays in the future.

The rest of the paper is organized as follows. We review previous work in Sec.2. The description of our testbed and experiment design is in Sec.3. In Sec.4 we present our results, and Sec.5 contains discussion and conclusions.

2 Previous work

Although the need for precise and detailed measurement of packet delays is recognized by the networking community, equipment constraints render it challenging, and the literature on this topic is scant. In particular, few researchers have access to high-precision (sub-microsecond precision) capture cards or to high performance routers representative of those deployed in Tier-1 ISP backbones.

Further, most previous work does not focus on ICMP delays per se, but rather on separating *forwarding* (that is, router transit) delays from queueing delays [9] or delays caused by network distance [10]. Bovy, *et al.*, estimated the forwarding delay of three office-class routers to be 224 μs per 100-byte packet per hop [10]. A wide variety of work in bandwidth estimation, much of it surveyed in [11] and [12], applies linear models of forwarding and queueing delays to the design of measurement tools. Discovery of Layer 2 devices by their delays is discussed in [13], [14].

Our related work on network spectroscopy focuses on identification of link-layer technologies [15] and OS fingerprinting by DNS updates [16]. Device fingerprinting by clock skews of TCP and ICMP timestamps is studied in [17].

Researchers from Sprint's Advanced Technology Laboratory (ATL) did several studies of instrumented operational routers in a setup close to ours [18], [19], [9], and support the claim that queueing delay in a well-provisioned network is small enough to effectively allow VOIP deployment [20].

A *Light Reading* test of Cisco, Juniper and Foundry measured forwarding delays at line rate (100% load) [21].

Govindan and Paxson [22] and Anagnostakis *et al.*[23] also study ICMP generation times, concluding that ICMP-based RTTs do not tend to include excessive (slow path) delays. Timing jitter in the network around routers complicates the attribution of these delays, but their value (0.1–0.3 ms) is comparable to those in [10] and to ours.

The goal of [23] is to infer link latencies and queueing from ICMP timestamp differences at both ends of a link (see also [6])[2]. The authors found routers (5 in 20 studied) with 95th percentiles of ICMP Timestamp delay around 10 ms; 2 had 95th percentiles at 80 ms. Remote link estimation is quite daunting in the face of such high uncertainty. For comparison, more than 99.6% of our TimeExceeded delays up to 9000 bytes are under 1 ms, except a few (0.4%) that are rate-limited by Juniper routers to incur approximately 10 ms delays.

Donnelly [25] and Mochalski *et al.*[26] demonstrate a piecewise linear size dependence for router/switch transit times, which shows a noticeable rate change at 512 bytes.

To the best of our knowledge, precision timestamping matching modern router speeds is available only with Dag cards from the Waikato group [27] and Endace [28]. The latest models (4.xx) can reach sub-microsecond accuracy when synchronized to GPS or CDMA [25] [29].

Some of the available studies use the now older model (3.xx) of Dag cards, with 5–6 μs precision [18] and 53-byte uncertainty with respect to the portion of the packet that is timestamped. Despite these limitations, the results obtained in [9], [18], and [19] have served as inspiration for this work.

3 Data collection

We collected our measurements in CAIDA's high-speed testbed [30] [12] which includes (Fig.1): two IBM eServers (running FreeBSD 4.8); a Dell Gigabit Ethernet switch; Juniper, Cisco and Foundry routers; an OC48 link between the Juniper and Cisco; and Gigabit Ethernet links between all other devices. The testbed's path MTU is 9000 bytes. We tap both links at the Cisco router (OC48 and GE) using NetOptics splitters, and capture packets sent to and from the router with Dag cards.[3]

[2] [24] suggests using traceroute delays for both purposes.

[3] Dag 4.23S for OC48 and Dag 4.3GE for GE link. Dag 4.23S timestamps first 4 bytes of each packet [25]. Our 4.3GE card had the following timestamping mechanism: "The framer will internally buffer arriving frames until either the end of frame is received, or the forwarding threshold (1540B) is reached. The consequence of this is that the timestamp will be captured at the end of frames $\leq$ 1540B in size, or 1540 bytes after the beginning of frames larger

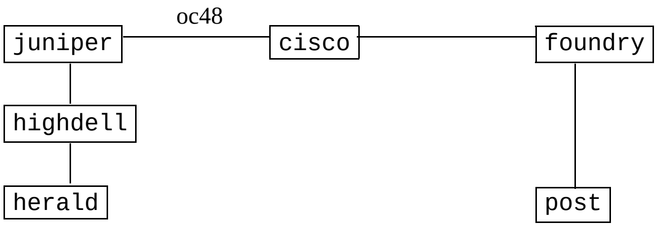

Fig. 1. Lab diagram. Equipment (clockwise): IBM eServer `herald`, Dell PowerConnect 5212 switch, Juniper M20 router, Cisco 12008 router, Foundry BigIron 8000 router/switch, IBM eServer `post`, Links: oc48 (Juniper-Cisco); Gigabit Ethernet (all other links). For details, see [12].

The Foundry router doubles as a 16-port switch that connects all equipment in the lab to the Internet and to CAIDA's production network via 100 M Ethernet.

We perform traceroutes on `herald` or `post`, and use CoralReef [32] utilities to capture, process, and extract delays from packets. A command line on `herald` of:

```
traceroute -q 4  -M 2  -m 3  -w 2  -P udp  -t 64  post  214
```

specifies series of 4 probes (q) to hops 2 (M) through 3 (m), using a timeout of 2 sec (w), UDP[4] (P), TOS of 64 (t) and packet size 214 bytes. Its output looks like (numbers from real data):

```
 2 cisco-oc48   0.221 ms  0.154 ms  0.254 ms  0.168 ms
 3 foundry       0.217 ms  0.226 ms  0.230 ms  0.227 ms
```

Our experiments combine UDP and ICMP traceroutes with 9 TOS values (0, 1, 2, 4, 8, 16, 32, 64, 128), and sizes 64-9000 bytes, for a total of 160866 (2*9*8937) traceroutes, each probing 2 hops with 4 packets at each hop. The router configuration guarantees that the return path for an ICMP packet is symmetric with the forward path.

Traceroute dynamics determine the intervals between probes in our experiments (Fig.2). We call a time lag between two successive packets targeting the same interface an *interprobe gap* (IPG). When traceroute probes one hop, it sends the next packet immediately after receiving an ICMP TimeExceeded for the previous packet. These probes succeed each other within a few hundred microseconds (under 1 ms). The next traceroute command will probe the same hop after an OS scheduling quantum (10 ms) and after probing a subsequent hop (several milliseconds); in that case, the probes are separated by 10-20 ms. When a TimeExceeded is not generated or is lost before the source host receives it (the loss is in fact very rare in our experiments) the traceroute waits for a 2-second timeout. This gap can affect the delay of the packet that follows, e.g. through route cache latency if the address has been flushed from the cache.

than 1540B. Any frames larger than ‘long’ (1518B default) will subsequenty be truncated to 1518B before being sent to the FPGA for timestamping. In 2.5.2 and subsequent releases, dagfour changes the threshold to one word (rather than 1540B). [...] The ‘packets’ at this stage are Ethernet frames. The preamble and sfd are discarded, so the frame consists of the mac addresses, vlan tag if present, Ethernet type field, Ethernet payload, and FCS.” [31]. We used the default value of “long,” which explains the slope changes in Figs.7 and 8. Firmware version 2.5.2 was released in October 2004 after our experiments were done.

[4] Recall that traceroute sends UDP or ICMP packets, but always gets back ICMP. Our data contains half UDP and half ICMP probes. The analysis presented here does not distinguish between UDP and ICMP probes, or between TOS values.

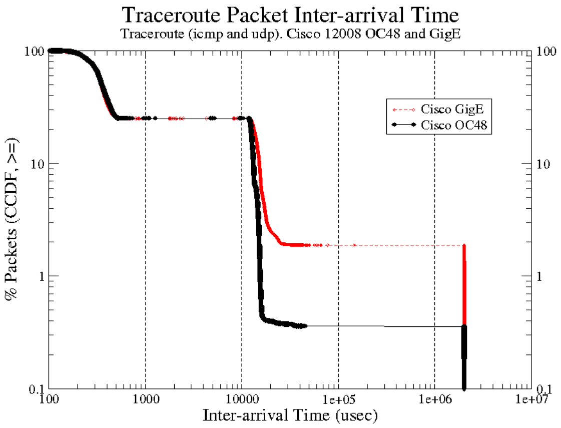

Fig. 2. Clustering of interprobe gaps for the Cisco router (OC48 and GE): microsecond range, 10–20 ms, 2 sec. The higher fraction of 2-sec gaps on the Cisco GE (upper curve) is caused by the Juniper not generating some ICMP messages.

Parameter scan. We walk the experiment design space (N_S packet sizes, N_P protocols, N_D destinations, N_T TOSes, etc.) using a pseudo-random scan. Scanning of other parameters (hop number, packets/hop) is a part of typical traceroute operation. We take the product of dimensions $m = N_S N_P N_D N_T \dots$ and find a prime $p > m$. Then we find a primitive root $r \bmod p$ near $\sqrt{p}$, and try all combinations of parameter values as follows. For experiment k, $1 \leq k \leq m$, we use $a_k = r^k \, \mathrm{mod} p$ in mixed-radix notation to get index S for size, P for protocol, D for destination etc:

$$S = a_k \mathrm{mod} N_S, \;\; P = [a_k/N_S] \mathrm{mod} N_P, \;\; D = [a_k/(N_S N_P)] \mathrm{mod} N_D, \;\; \text{etc.} \;\; (a_k \leq m)$$

Example. For two packet sizes ($N_S = 2$) and two protocols ($N_P = 2$), $m = N_S N_P = 4$ and $p = 5$; $r = 3$ is a possible choice of a primitive root. Combinations of packet size (e.g. (40, 1500) indexed by (0,1)) and protocol ((UDP, ICMP) indexed by (0,1)) follow each other in sequence[5] $(3^1, 3^2, 3^3, 3^4) \mathrm{mod} 5 = (3, 4, 2, 1) = (11, 00, 10, 01)_2$, where 11 corresponds to (ICMP, 1500), and so on.

This approach, inspired by turbo codes [33] and Monte-Carlo integration techniques, is robust against outages, whether at the beginning (Dag cards warming up) or at the end (too small capture interval, disk space). All parameter values appear close to the start of experiment (as opposed to with a lexicographic scan), which allows us to debug problems with each dimension or value, e.g. too high chance of a timeout.

Table 1 presents a description of the data in terms of destinations, experiment duration, number of traceroutes and number of probes (packets). The second half of the table is a breakdown of the probes by interprobe gap (IPG). The longer duration of the second (PCJ) experiment is due to a higher level of ICMP non-generation on Juniper (12140 or 2% of all probes) which results in more occurrences of the 2-sec timeouts. This extra 10K (12140-2310) of timeouts increases the experiment duration by about

[5] In this special case, one can read parameters from the two rightmost bits of $r^k \mathrm{mod} p$.

5.5 hours. In addition, Juniper's generation bitrate of TimeExceeded (at 8 ns/bit) is the slowest of all three routers (Table 2). ICMP bitrate limiting causes many packets in the 7000–9000 byte range (73K or 11%) to arrive more than 1 ms later than the previous probe. This lag applies to packets 2–4. Packet 1 is always delayed by an OS scheduling quantum of 10 ms, which explains the large number of packets (about 25% of the total) in the 10–100 ms bin. The drop rate (non-generation) for the Foundry is under 0.4%, and the Cisco returns all 643464 probes, i.e. has 0% drop rate.

Table 1. Experimental data and interprobe gaps

Code	Source	Destination	Date	Start	End	Traceroutes	Packets sent
HCF	herald	Cisco, Foundry	2004-09-10	00:00	02:00	160866	1287 K
PCJ	post	Cisco, Juniper	2004-09-12	00:30	08:00	160866	1287 K

Code	Source	Dest.	i/face	IPG<1ms	1–10ms	10–100ms	0.1–1s	IPG>1s	Total
HCF	herald	Cisco	OC48	482546	20	158587	0	2310	643463
HCF	herald	Foundry	GE	477557	539	160747	0	2310	641153
PCJ	post	Cisco	GE	482570	19	148733	1	12140	643463
PCJ	post	Juniper	OC48	389211	72793	157178	1	12140	631323

4 Results

We define router delay as the time elapsing from the moment the first byte of a packet enters the router to the moment the first byte of the reply exits the router. This quantity can be easily measured with a capture card that timestamps at the beginning of packets.[6] The Dag 4.23S card used on the OC48 link does exactly that [25] [31] — it timestamps the first 4 bytes of a Layer 2 frame. However, instead of timestamping the first bytes, the Dag 4.3GE card used on the GE link timestamped the byte min(x,1540), where x is the Ethernet frame size.[7] Therefore, a packet starts entering the router min$(x, 1540) \cdot 8/C$ (where $C = 1$ Gbps) seconds earlier than the starting timestamp reported by the Dag 4.3GE card and starts leaving the router $m \cdot 8/C$ seconds earlier than the ending timestamp, where m is the size of the framed ICMP TimeExceeded message (which has 56 IP bytes). The difference in Dag 4.3GE timestamps is thus (min$(x, 1540) - m) \cdot 8/C$ seconds smaller than the actual router delay. Since m is 74 or 78 (56 IP bytes plus 18 or 22 bytes depending on the use of VLAN tags), the timestamp difference is about 11.7 μs less than the actual delay for $x = 1540$ bytes and larger frames. All measurements we report for the GE link are the *unadjusted* router delays computed directly from the timestamps returned by the Dag 4.3GE card. This has no effect on the basic conclusions.

Table 2 provides a lower bound with size dependence parameters from equation $d = ax + b$: a (slope) and b (intercept) of TimeExceeded delay. We apply the $O(N)$ linear programming (LP) algorithm of [35] (cf. [36], [17]) to delays observed at the Cisco and Juniper OC48 interfaces for all packet sizes, and to those at the Cisco and Foundry GE interfaces only for range over 1500 bytes.

[6] The alternative approach of taking the timestamp at the end of a packet [9] can be harder to use for determining router delay because a packet's size depends on its content with Sonet [34].

[7] This behavior was changed by Endace in October 2004 [31]; see footnote in Sec.3.

The slopes is Tab.2 are in ns/bit (not μs/byte), to match the GE rate, 1 ns/bit. The intercept at 0 and the values of $ax + b$ at three packet sizes (x = 40, 1500, and 9000 bytes) are the minimum delays including deserialization (but not serialization).

The only router/probe type with ICMP generation rate equal to link rate is the Foundry TimeExceeded; others have smaller or larger slopes. Note that small slopes a can trick variable packet size (VPS) tools [7] into capacity overestimation, whereas slower-than-link rates (higher values of a) can result in underestimation. Both effects were observed in our lab with pathchar.[8] This situation is similar to the underestimation caused by extra serialization at Layer 2 switches invisible to traceroute [14].

Table 2. Linear fit of lower bound on TimeExceeded delay measured by Dag timestamp difference. Numbers in parentheses are extrapolations.

Router	**Slope**	**Lower bound (μs)**			
	(ns/bit)	0	40B	1500B	9000B
Cisco OC48, all	0.732	18.41	18.64	27.19	71.10
Juniper OC48, all	8.091	122.63	125.22	219.72	705.18
Cisco GE* > 1500	1.313	(6.66)	(7.08)	22.42	101.19
Foundry GE * > 1500	0.996	(15.90)	(16.22)	27.85	87.59

* Note that actual delays for GE link are about 11.7 μs higher than shown in the table. Packets under 1500 bytes (not shown) have apparent GE slopes about 1 ns/bit less than those in the table.

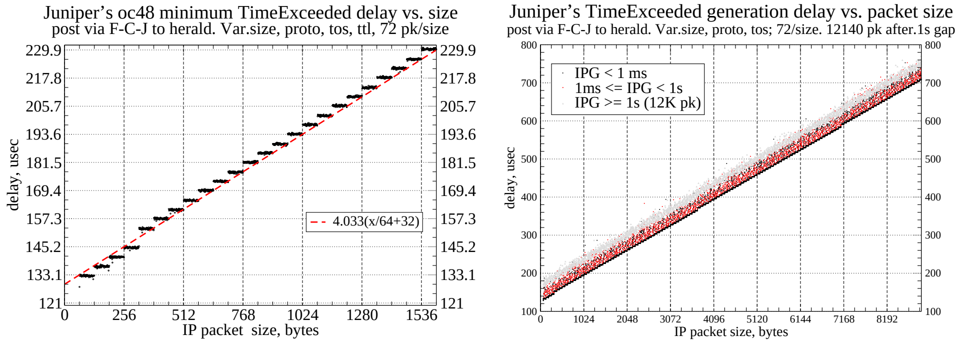

Fig. 3. (a) Minimum TimeExceeded delay from Juniper (left) with a staircase of 64-byte segments, 4 μs steps and an 8-μs jump at 320 bytes; (b) TimeExceeded delay from Juniper (right) showing about 30 μs of extra delay for an interprobe gap (IPG) of 2 sec. Three bands of delays result from the three ranges of interprobe gaps: upper light-colored band for IPG $\geq$ 1s, medium dark band for IPG between 1ms and 1s, and lower dark band for IPG $<$ 1ms.

Delays through the Juniper router are special in several respects (Fig.3). The minimum delay of the TimeExceeded packets grows stepwise by approximately 4.033 μs per 64-byte cell for sizes 64–320 bytes: $d = 4.033\lceil x/64 + 31\rceil\mu$s where $\lceil x\rceil$ is the smallest integer greater or equal to x. This formula is similar to that for ATM delays from [15] (cf.[14]) although the fixed cost (which for 64-byte packets is 128 μs, an equivalent of

[8] For example, pathchar measures 114 Mbps for OC48 and 101 Mbps for GE connection at Juniper. (Note that 114 Mbps $\approx 1/8.091$ bits/ns.)

almost 40 KB at the OC48 wire speed) is much higher than ATM's encapsulation cost. This cell rate would result in an average bitrate of 7.877 ns/bit, or 127 Mbps. However, the experimental curve jumps by an extra cell's worth of delay right after 320, 3712 and 7104 bytes (which are separated by 3392 bytes, or 53 64-byte cells). As a consequence, the slope in the linear programming-based lower bound is somewhat higher. That is, 54 cells worth of delay per 53 cells of size equals 8.026 ns/bit, but the LP estimate from Table 2 is 8.091 ns/bit, which may imply a 0.8% error in 4.033 μs cell time. Fig.3a shows a close-up for packets under 1600 bytes. The staircase of minimum delays starts under the line $y = 4.033x/64 + 32$ (recall that $\lceil x \rceil < x + 1$) but crosses over the line between 320 and 321 bytes. This size-delay dependence is obviously nonlinear. The 8-μs jump at 320 bytes and the accumulated discrepancy with the straight line (12 μs over 9000 bytes interval) can potentially be measured by traceroute-like tools, even though individual 4μs-steps may be hard to discern from network noise.

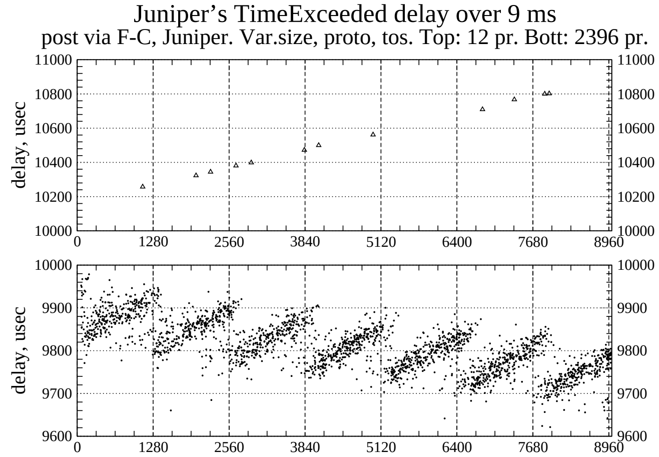

Fig. 4. TimeExceeded message delay from Juniper OC48. Values over 10 ms (top) and 9–10 ms (bottom). Values between 9–10 ms reveal unusual size dependence of ICMP TimeExceeded generation delay when ICMP is rate-limited to 100 packet per second (one packet in 10 ms).

The Juniper router also delayed widely spaced (interprobe gap of 2s) packets by about 30 μs compared with closely spaced packets of the same size. This delay could be due to route cache flushing. The delay pattern in Fig.4 (bottom) which holds for 2400 (0.4%) probes rate-limited to 10 ms (packets 2–4 in some traceroutes) has a prominent negative trend that could potentially be used for fingerprinting.

Figs. 5 and 6 show the dependence of ICMP delays on packet size for the Cisco OC48 interface, separated into three sets by interprobe gap (time between traceroute packets): under 1 ms, 1ms–1s, over 1 s. The actual distibution of the longer lulls clusters around 10 ms (kernel scheduling quantum) and 2 sec (traceroute timeout), both described in Sec.3 (Fig.2). Probes delayed by 10–20 ms span a wider range of ξ (reflected in the width of the middle strip in Fig. 6) than probes sent immediately after the previous probe, but at the same time many of them are close to the linear lower bound (which coincides with the bottom of the strip). On the other hand, probes sent after the 2-second timeout always encounter an extra delay of about 20 μs. The banding of ξ here and in Figs. 7, 8, and 9 may be due to route cache flushing and other state lost after certain time intervals. We observed similar dependence on the duration of the lull between the previous packet and the current probe for Juniper (Fig.3b) and Foundry routers (Fig.8). To give an idea of the average density of points in these bands, Fig.9 shows a histogram of residual delay ξ, i.e. the delay less the lower bound of delay shown in Table 2 for sizes below and above 1500 bytes (partial Radon transform [15]). Note that this summary histogram suggests (but does not prove) the stationarity of ξ with respect to packet size. While this stationarity is typically assumed, our preliminary results show that it at best only approximately holds.

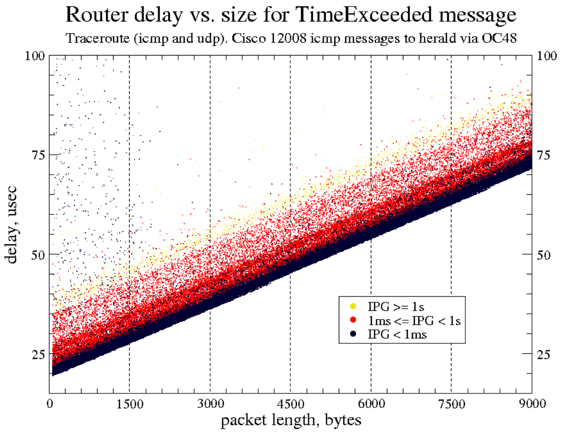

Fig. 5. TimeExceeded message delay from Cisco OC48. Compare with Fig.6 where each cluster of interprobe gaps is in its own panel.

A common assumption in network research is that an idle router processes packets with minimum possible delay [37]. Our experimental setup guarantees that no cross-traffic is present and that routers process probes one at a time. Table 3 presents statistics (average, 95%, 99% and maximum for the whole datasets without groupings by interprobe gap) for the residual delay ξ, i.e. ICMP generation time in excess of linear lower bound $ax + b$ (where Table 2 shows the slope a and intercept b).

Table 3. Statistics of residual delay ξ for ICMP TimeExceeded (generation time in excess of delays attributed to packet size) on Cisco and Foundry's GE line cards. Bin size 0.238 μs (i.e. 2^{-22} s); closest percentile selected.

Router	Packet	Packet	Delay (μs)			
	Size	Count	avg	95%	99%	max
Cisco	$\leq$ 1500	103463	2.598	6.199	20.504	296.593
Cisco	$>$ 1500	540000	2.252	5.484	18.835	281.096
Foundry	$\leq$ 1500	103075	4.406	3.338	31.233	1537.800
Foundry	$>$ 1500	538078	4.996	3.815	31.948	1492.500

We can summarize Table 3 as follows: Cisco and Foundry GE interfaces process TimeExceeded with no more that 6 μs of extra delay (over the size-dependent lower bound)

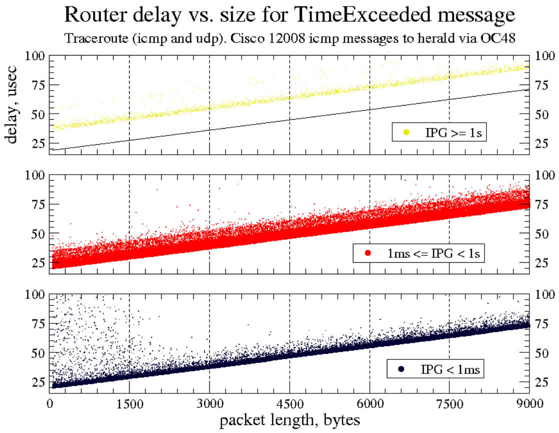

Fig. 6. TimeExceeded message delay from Cisco OC48. Panels from top to bottom: delay for interprobe gap of over 1 sec, 1 ms–1 sec and under 1 ms. The position of the curve in the top panel reflects about 20 μs of extra delay (presumably route cache warm-up) beyond the lower bound of all delays, which is indicated by the solid line. The bottom panel shows some scattering of delays (possibly from rate limiting) for closely spaced packets under 3000 bytes.

in 95% of cases; however, for 1% of packets the extra delay is between 20 and 300 μs on the Cisco and 30–1500 μs on the Foundry. These values of ξ should be remotely measurable. On the other hand, the statistics of ξ are close to each other for packets with sizes under and over 1500 bytes, which is more in line with common wisdom.

5 Discussion, conclusions, future work

Our paper is the first study of router delays in the full packet size range 40–9000 bytes. Getting precise data for the whole range is a pressing issue since providers like Abilene, Geant, and Switch are already supporting 9000-byte transparent paths, and since the global Internet transition toward these larger packet sizes is only a matter of time.

We demonstrated that a linear model of ICMP delay is an approximation (like Newtonian mechanics) that breaks down when cell-based processing is involved, such as with the Juniper router. [9] Designers of bandwidth estimation and other measurement tools [12] must be aware of this reality.

We find that (for all packet sizes) delays above the minimum are not necessarily due to queueing. For example we observed that Juniper delays some closely spaced traceroute packets by 9–10 ms (Fig.4). However, our measurements of Cisco and Foundry's GE interfaces (Table 3) show that for most (95%) probes the extra delay is within a few microseconds, and it is within 300 μs for Cisco (and 1.5 ms for Foundry) over the whole sample, which is negligible for (although measurable by) many applications.

[9] For Juniper, the delay rates and values for EchoReply and PortUnreachable are discontinuous at certain packet sizes; phenomena of that kind were also observed at our Cisco router.

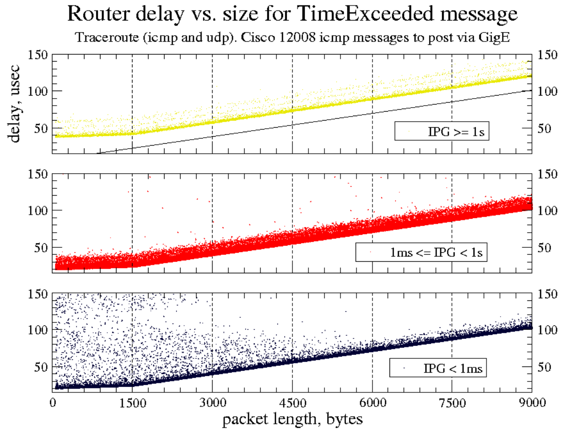

Fig. 7. TimeExceeded message delay from Cisco GE (NOTE: The piecewise nonlinearity at 1500B is an artifact of the Dag 4.3GE card). Panels from top to bottom: delay for interprobe gap of over 1 sec, 1 ms–1 sec, and under 1 ms. Unlike the Foundry data, the slope is greater that 1 ns/bit, reflecting rate limiting of ICMP replies by about 30%. The position of the curve in the top panel reflects about 20 μs of extra delay (presumably route cache warm-up) beyond the lower bound of all delays indicated by the dashed line. The bottom panel shows some scattering of delays (possibly from rate limiting) for closely spaced packets under 4500 bytes.

Surprisingly, we found that the ICMP rate can differ by a factor of 20 from the link rate, depending on router and ICMP type. This ambiguity suggests that capacity estimates by ICMP-based tools [7] [38] [39] may need to make heavy use of router and even interface fingerprinting, rather than just filtering and fitting as if ‘all RTT data are created equal’. Our results complement previous insights in VPS tools issues [14].

We found that Juniper’s TimeExceeded processing is based on 64-byte cells (Fig.3a). We plan to investigate whether the 48-byte cell[10] granularity of the Cisco documented in [40] is present in our data.

Our analysis shows that ICMP delay can depend on packet size and header fields in various non-intuitive ways, including:

- stepwise growth, e.g. each 64 bytes with occasional jumps (Fig.3)
- drops causing an overall decreasing trend vs. packet size (Fig. 4)
- internal tasks that postpone packet scheduling by fixed delays (clustering in distinct “bands”) on an absolutely empty device (Fig. 8, 9)
- warming up caches can cause significant (20-30 μs) extra latency for widely spaced probes, e.g., an interprobe gap of seconds (Fig. 6, 5, 7, 8), which explains the mystery of first probe in traceroute and ping always having higher RTT.

Table 4 summarizes our main results and lists two cases of linearity of message generation delay with respect to packet size (approximately linear, stepwise linear with jumps) observed for the three router types studied. In contrast with prevalent assumptions used by some rate estimation tools, only one of our studied routers has a TimeExceeded generation rate equal to the line rate of the inbound link. One router has an

[10] “The Fabric Interface ASIC is set up to segment the packet into 48-byte cells.” [40].

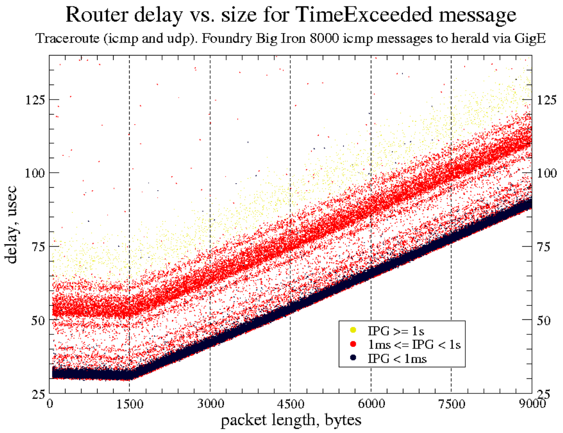

Fig. 8. TimeExceeded message delay from Foundry (NOTE: The piecewise nonlinearity at 1500B is an artifact of the Dag 4.3GE card). The slope is close to 1 ns/bit, as assumed by VPS tools.

ICMP rate that is 20 times slower than its line rate (the ratio of generation rate to line rate is 0.05, Table 4). Another router slows down ICMP by 80% on the OC48 interface and by 30% on the GE interface. These properties can facilitate remote device/link fingerprinting. Taken together, our results indicate surprisingly different attitudes of router vendors (from restrictive to receptive to acceptive) with regard to ICMP Time Exceeded messages. Our work in progress suggests that many of these attitudes apply to other ICMP messages too.

Areas for further investigation include confirming details on the phenomena mentioned above, as well as forwarding delays, payload dependent delays, cross-traffic effects, rate estimates based on optimization technique of [15], and independence tests.

Table 4. Observed behavior of routers responding with ICMP TimeExceeded messages

Property	**Juniper OC48**	**Cisco OC48**	**Cisco GE**	**Foundry GE**
Message generation linearity	steps w.jumps	approx.linear	approx.linear	approx.linear
Min.latency, all packets $\geq$ 64B	128 μs	19.4 μs	19.4 μs	29.2 μs
Generation rate/Line rate, $\leq$ 1500B	0.05	1.37	3.1	negative
ICMP non-generation rate	2%	0%	0%	0.4%

6 Acknowledgements

Dan Andersen, Brendan White, Grant Duvall, Margaret Murray, and Kevin Walsh created the lab used in this study. Ken Keys helped with the Coral software. Stephen Donnelly clarified Dag timestamping. Yoshi Kohno provided the linear programming code. Thanks to the PAM reviewers for their comments, all of which we incorporated in the text. Thanks also to Allen Porter, Dave Berlin, Niheer Patel, Xin Cai, Andrey Shapiro, and Tin Tran for their useful feedback on this report.

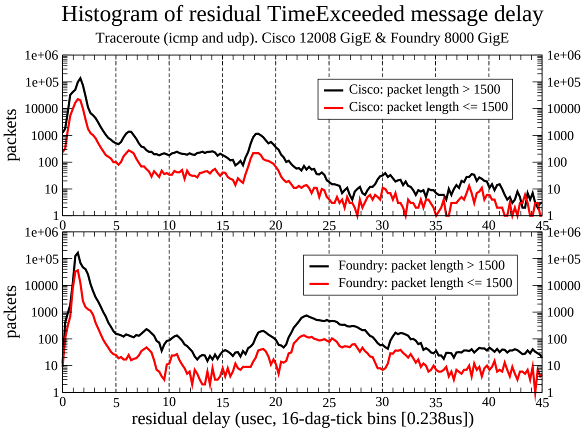

Fig. 9. Histogram of residual TimeExceeded delay. Positions of the maxima are similar for packets under and over 1500 bytes, suggesting that residual delay ξ is not strongly dependent on size.

References

1. Pasztor, A., Veitch, D.: The packet size dependence of packet pair like methods. In: IWQoS. (2002)
2. Jain, M., Dovrolis, C.: End-to-end available bandwidth: measurement methodology, dynamics, and relation with TCP throughput. In: Sigcomm. (2002)
3. Katti, S., Katabi, D., Blake, C., Kohler, E., Strauss, J.: Multiq: Automated detection of multiple bottlenecks along a path. In: IMC. (2004)
4. Spring, N., Mahajan, R., Wetherall, D.: Measuring ISP topologies with Rocketfuel. In: Sigcomm. (2002)
5. Spring, N., Wetherall, D., Anderson, T.: Reverse engineering the Internet. In: HotNets. (2003)
6. Mahajan, R., Spring, N., Wetherall, D., Anderson, T.: User-level Internet path diagnosis. In: SOSP. (2003)
7. Jacobson, V.: pathchar - a tool to infer characteristics of Internet paths (1997) ftp.ee.lbl.gov/pathchar.
8. Broido, A., claffy, k.: Internet Topology: connectivity of IP graphs. In: SPIE, vol.4526. (2001)
9. Hohn, N., Veitch, D., Papagiannaki, K., Diot, C.: Bridging router performance and queueing theory. In: Sigmetrics. (2004)
10. Bovy, C.J., Mertodimedjo, H.T., Hooghiemstra, G., Uijtervaal, H., van Mieghem, P.: Analysis of end-to-end delay measurements in Internet. In: PAM. (2002)
11. Prasad, R.S., Murray, M., Dovrolis, C., claffy, k.: Bandwidth estimation: metrics, measurements, techniques and tools. In: IEEE Network. (2004)
12. Sriram, A., Murray, M., Hyun, Y., Brownlee, N., Broido, A., Fomenkov, M., claffy, k.: Comparison of public end-to-end bandwidth estimation tools on high-speed links. In: PAM. (2005)
13. Pasztor, A., Veitch, D.: Active probing using packet quartets. In: IMW. (2002)

14. Prasad, R., Dovrolis, C., Mah, B.: The effect of store-and-forward devices on per-hop capacity estimation. In: Infocom. (2003)
15. Broido, A., King, R., Nemeth, E., claffy, k.: Radon spectroscopy of inter-packet delay. In: IEEE High-speed networking workshop. (2003)
16. Broido, A., Nemeth, E., claffy, k.: Spectroscopy of DNS update traffic. In: Sigmetrics. (2003)
17. Kohno, Y., Broido, A., claffy, k.: Remote physical device fingerprinting. In: IEEE Symposium on Security and Privacy. (2005)
18. Papagiannaki, K., Moon, S., Fraleigh, C., Thiran, P., Tobagi, F., Diot, C.: Analysis of measured single-hop delay from an operational network. In: Infocom. (2002)
19. Choi, B.Y., Moon, S., Zhang, Z.L., Papagiannaki, K., Diot, C.: Analysis of point-to-point packet delay in an operational network. In: Infocom. (2004)
20. Fraleigh, C., Tobagi, F., Diot, C.: Provisioning IP backbone networks to support latency sensitive traffic. In: Infocom. (2003)
21. Newman, D., Chagnot, G., Perser, J.: The internet core routing test: Complete results (2001) www.lightreading.com/document.asp?doc_id=6411.
22. Govindan, R., Paxson, V.: Estimating router ICMP generation delays. In: PAM. (2002)
23. Anagnostakis, K., Greenwald, M., Ryger, R.: cing: Measuring network-internal delays. In: Infocom. (2003)
24. Akela, A., Seshan, S., Shaikh, A.: An empirical evaluation of wide-area internet bottlenecks. In: IMC. (2003)
25. Donnelly, S.: High precision timing in passive measurements of data networks (2002) Ph.D. thesis, University of Waikato, Hamilton, New Zealand.
26. Mochalski, K., Micheel, J., Donnelly, S.: Packet delay and loss at the Auckland Internet access path. In: PAM. (2002)
27. Graham, I.D., Pearson, M., Martens, J., Donnelly, S.: Dag - A cell capture board for ATM measurement systems (1997) www.cs.waikato.ac.nz /Pub/Html/ATMDag/dag.html.
28. Endace: Measurement Systems (2004) www.endace.com.
29. Micheel, J., Donnelly, S., Graham, I.: Precision timestamping of network packets. In: IMW. (2001)
30. CAIDA: Bandwidth estimation project (2004) www.caida.org/projects/bwest.
31. Donnelly, S.: Private communication (2005)
32. Keys, K., Moore, D., Koga, R., Lagache, E., Tesch, M., claffy, k.: The architecture of CoralReef: Internet Traffic monitoring software suite. In: PAM. (2001)
33. Berrou, C., Glavieux, A., Thitimajshima, P.: Near Shannon limit error-correcting coding and encoding: turbo codes. In: IEEE Int'l Conference on Conmmunications. (1993)
34. Simpson, W.: PPP in HDLC-like framing, RFC 1662. (1994)
35. Moon, S.B., Skelly, P., Towsley, D.: Estimation and removal of clock skew from network delay measurements (1998) Tech.Rep.98-43, UMass Amherst (Infocom 1999).
36. Graham, R.L.: An efficient algorithm for determining a convex hull of a finite planar set. In: Info.Proc.Lett. 1. (1972)
37. Ribeiro, V., Riedi, R., Baraniuk, R., Navratil, J., Cottrell, L.: pathChirp: Efficient Available Bandwidth Estimation for Network Paths. In: PAM. (2003)
38. Mah, B.: pchar: a tool for measuring Internet path characteristics (1999) www.kitchenlab.org/www/bmah/Software/pchar.
39. Downey, A.B.: Using pathchar to estimate Internet link characteristics. In: Sigcomm. (1999)
40. Cisco: How to read the output of the show controller frfab / tofab queue commands on a Cisco 12000 Series Internet Router. Document ID 18002 (2004) www.cisco.com.