USENIX Association

Proceedings of the LISA 2001 15th Systems Administration Conference

San Diego, California, USA
December 2–7, 2001

DNS Root/gTLD Performance Measurement

Nevil Brownlee – The University of Auckland, New Zealand and CAIDA, SDSC, UC San Diego
kc claffy – CAIDA, SDSC UC San Diego
Evi Nemeth – University of Colorado and CAIDA, SDSC, UC San Diego

ABSTRACT

The Internet Domain Name System (DNS) is an essential part of the Internet infrastructure. Each web site or email lookup involves traversing a tree-structured distributed database to complete the mapping from a hostname to an IP address. The root and top level domain (TLD) nameservers form the highest level of authority over the Internet naming hierarchy, and are thus potentially involved in reaching any and every URL or email address we seek. We use passive measurements to analyze performance of these critical nameservers from a client network's viewpoint.[1]

We use NeTraMet meters on a university campus to take passive measurements of DNS response time, request loss rate and request load to the root and gTLD (generic top level domain, e.g., .com, .net, .org) servers.

From these measurements we produce strip charts that are useful for day-to-day monitoring of one's Internet connectivity, since they reveal changes in network behavior on paths between one's local network and the global servers without the need to actively inject traffic into the network. We are developing a monitoring tool to produce such plots in near real time.

Introduction

DNS, the Domain Name System, is responsible for translating between hostnames used by people and corresponding IP addresses needed by software. The data for this mapping is stored in a tree-structured distributed database where each nameserver is authoritatively (responsible) for a portion of the naming hierarchy.

The DNS protocol [1] is a request/response protocol based on UDP transport. BIND [2], the Berkeley Internet Name Domain System, is the reference implementation used by most sites. Local nameservers contact root nameservers and then traverse the DNS tree until their query arrives at a server that has the answer. Root servers provide referrals to nameservers for country-code domains (e.g., .au, .uk, .us), and generic top-level domains (gTLDs, e.g., .com, .net, .org). A referral is basically an answer that says "I do not know the address of yahoo.com, but here is the address of the .com servers who *will* have that information."

BIND has a built-in load balancing mechanism when a query results in more than one answer, as happens for a query for the root zone (13 answers) or gTLD zones (11 answers). When BIND gets a response with multiple answers, it keeps track of the Round Trip Time (RTT) to each of the servers in the answer and sorts them into bands. BIND will then round robin among the servers in the band with the lowest RTT, but also periodically reduces the RTT stored for each server in other bands. These far away servers will thus eventually be in the closest band, BIND will then query and resort them based on a more recent RTT. This technique causes BIND to usually choose the servers that are nearest in terms of latency, and spread the load across them, but also to re-calibrate itself in case a server's performance was anomalous rather than representative.

As well as load balancing, BIND caches replies it receives from other nameservers. Each answer carries a Time To Live (TTL) value, telling the local nameserver how long it may locally cache that answer. For frequently used domain names BIND will usually have a cached answer. As well as improving response time to the user, caching dramatically reduces the load on upstream nameservers in the tree. In other words, caching makes the DNS scale.

Figure 1 shows the location of the global root and gTLD nameservers. Each has a one-letter name, i.e., A, B, . . ., M, and is identified by city. The servers are not evenly spread throughout the world – they are clustered on the east and west coasts of the US. For example, the A, C, D, G, H and J roots and the A and G gTLDs are all located in the vicinity of Washington, DC. Similarly, there are four roots and four gTLDs on the US West Coast, near San Francisco and Los Angeles. The roots are run by individual organizations, the gTLDs are all administered by Network Solutions.

[1] Support for this work is provided by DARPA NGI Contract N66001-98-2-8922, NSF Award NCR-9711092 'CAIDA: Cooperative Association for Internet Data Analysis,' and The University of Auckland.

The root and gTLD nameservers are crucial to the Internet infrastructure. Until mid-2000 the root servers also served the gTLDs. As the growth of the Internet increased the workload on the roots, the .com, .net, and .org domains were moved off the root system onto a separate layer of 'global top level domain' (gTLD) servers run by Network Solutions. There are 11 of these gTLD servers. While the hardware and operating systems of the root servers is architecturally diverse, with several manufacturers and operating system vendors represented, all 11 gTLD servers are identical IBM AIX machines, distributed around the world.

We use a traffic metering tool called *NeTraMet* [3] to examine behavior of the root and gTLD servers from a client site's perspective. We run *NeTraMet* on top of an OC12 (622 Mb/s) link monitor, CoralReef [4], that sees university traffic via an optical splitter on the wide area circuits connecting the university to the Internet. We measure request rates, response times (RTT) by matching response and request packets, and request loss rates.

A recent paper on Internet performance includes DNS name resolution latency measurements. Huitema & Weerhandi [5] found that the end-to-end latency of name lookup exceeded two seconds in 29% of the cases. Our measurements do not show nearly this amount of delay, however they are not end-to-end measurements, but rather edge of campus to root servers or gTLD servers and back. Perhaps the discrepancy in the observations derives from a slow or congested Internet connection at the point of their measurements or is due to efficient caching on our campus. At the time of Huitema & Weerhandi's measurements, the root servers also served the gTLD domains and thus may have been overloaded.

In the next section, we describe our measurement methodology, and then discuss the client-side measurement results.

Measurement Methodology

Our measurements are passive, which means we observe the traffic flowing by, rather than actively injecting any traffic into the network. As it observes DNS packets, our traffic meter performs data reduction, such as computing response times. The meter writes the reduced data to 'flow data' files for later analysis.

Our campus network is connected to the commodity Internet and to three research/academic networks. In May 2000 we installed one OC3 (155 Mbps) traffic meter, our *commodity* meter, to monitor the commodity Internet link. Our campus network topology is such that there is no single point where a traffic meter can see traffic for the entire campus on all four external links. Instead, in January 2001, we placed a second OC3 meter at a boundary point within the campus network where it could see traffic for the inner part of the network, including a large fraction of the traffic to the commodity link and two of the three research/academic links. We call this our *edu* meter. In July 2001 our network configuration changed. We responded by reconfiguring our *edu* meter to monitor an OC12 link at a point in the new topology where it could see traffic to and from all four external links. Initial measurements in June and November 2000 used our *commodity* meter; the January 2001 measurements use both meters and the July 2001 measurements use only our OC12 *edu* meter.

We also have data measured at a site in New Zealand; unfortunately the request rate there was too low to provide reliable estimates of response times or request loss rates for the global root servers. Even so, our New Zealand data correlates well with that collected at San Diego.

Our traffic meter [3, 6, 7, 8] is an open source implementation of the IETF's Realtime Traffic Flow Measurement (RTFM) architecture for network traffic

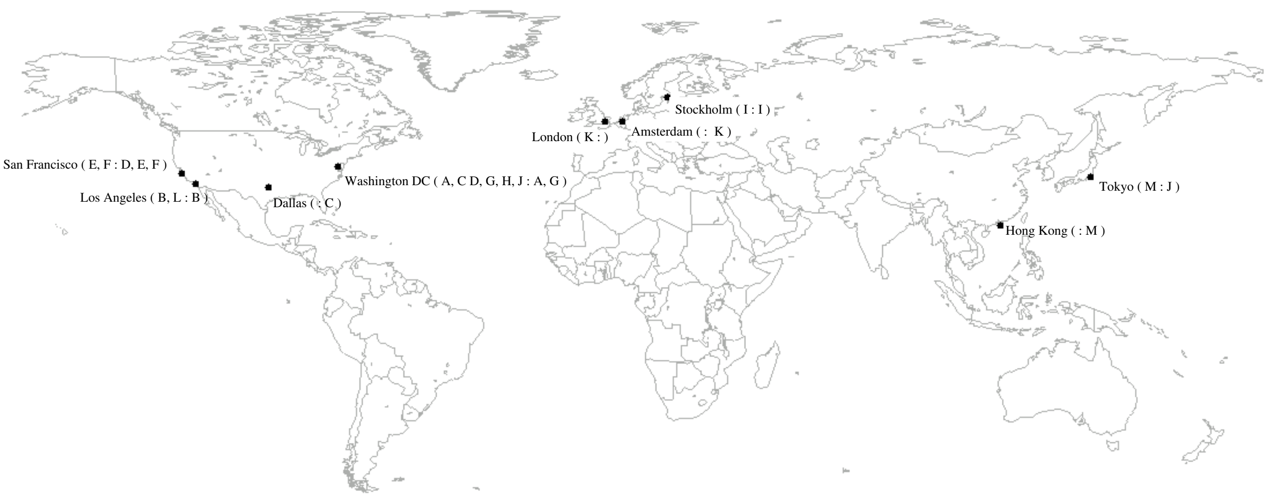

Figure 1: Location of root and gTLD servers. Each city lists servers (root: gTLD) using their one-letter names. Note the uneven geographical server distribution, with high concentrations on the US East and West Coasts.

flow measurement [9, 10, 11]. It is a highly configurable, passive, real-time link measurement tool, and has been extended to use the CoralReef library [4] to read packet headers from either a live network or from a trace file.

We used the traffic meter to measure response times for the global DNS root and gTLD nameservers by configuring it to capture DNS request packets and their corresponding response packets. The meter maintains queues of packets for each source-destination address pair and uses the ID field of the DNS header to match each incoming response with a queued request.

We also configured the meter to measure the number of 'unanswered' DNS requests, i.e., requests for which our meter did not capture a response. We use 50 bins for our distributions, with a logarithmic response-time range of 7 to 700 ms; response times above 700 ms are counted in a single overflow bin. Requests are considered unanswered if they do not receive a response within 10 times the highest bin value, i.e., 7 seconds.

We filter our captured data to include nameserver traffic travelling in both directions. To ascertain the degree of asymmetric routing affecting our global DNS traffic, we configured our meters to count DNS

- requests that receive a matching response. We use these to compute response times, which include the server response time as well as the network travel times.
- 'unanswered' requests. We compare these with the matched requests to produce plots of the request loss rate.
- 'unrequested' replies, i.e., response packets for which our meter did not previously capture a corresponding request.

DNS requests may be unanswered for several reasons:

- A request may come from a host with an invalid IP address, for example one from 'private' address space [12]; a reply cannot be routed back to such a host. Our meter is configured to ignore requests from addresses that do not belong to prefixes announced by our campus.
- A query may be badly formed. For example DNS queries should only contain a single question; queries with multiple questions violate the DNS protocol specification; the root servers will drop such requests. Some root servers see a

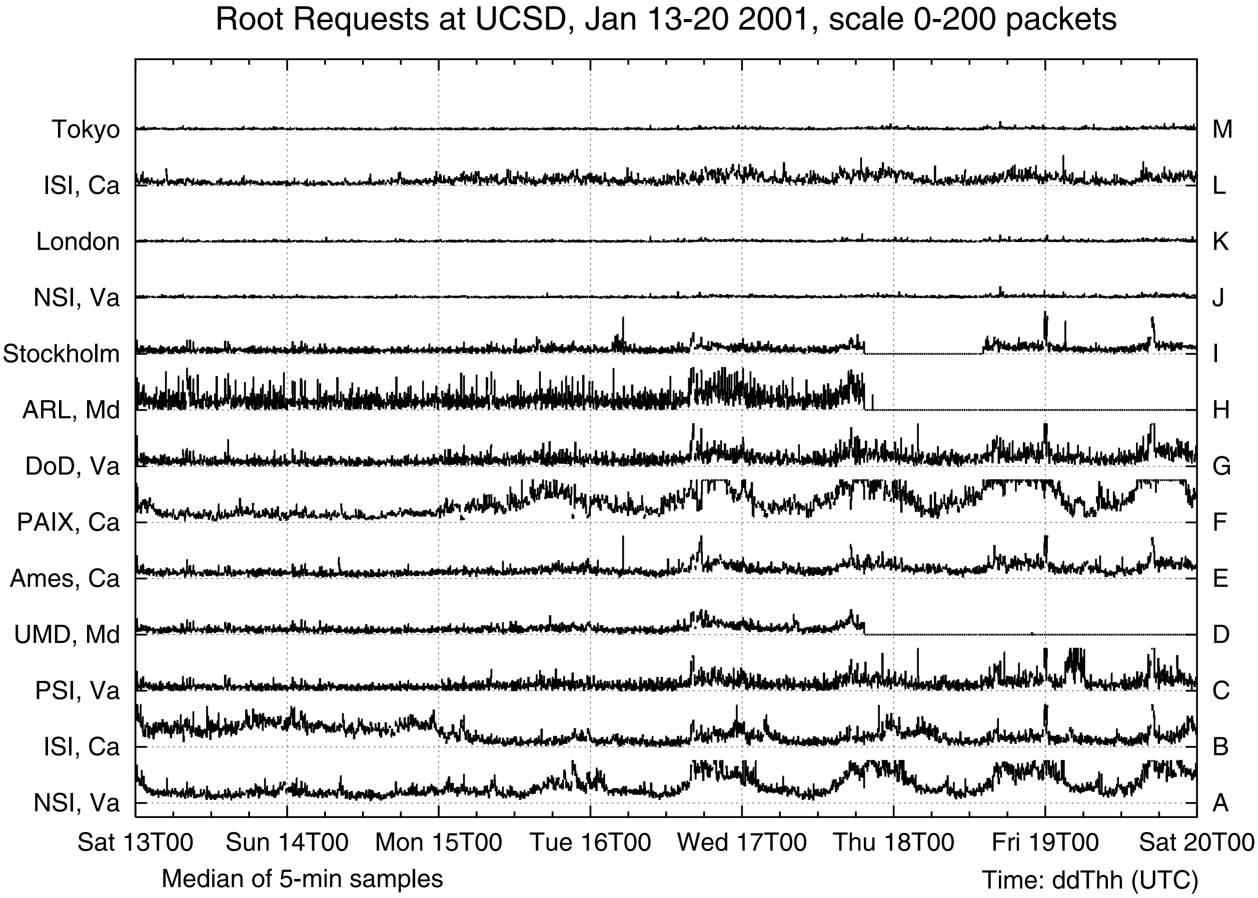

Figure 2: DNS requests to root servers reflect our local BIND's view of the servers. Here BIND is favoring A and F roots (we see clear diurnal variations), and sending few requests to J, K and M. BIND stops sending requests to D, H and I at 2000 on Wed 17 Jan, indicating that connectivity to them has been lost.

significant number of queries whose DNS header indicates 256 queries in the packet, when in fact there is only one – a Microsoft byte-order bug affecting the header field [13].

- Asymmetric routing may cause a request and its response to travel via different networks, so that a meter may see only one of these two packets. As indicated above, our meters are placed and configured so as to minimize this problem.
- A packet may be dropped on the way to the nameserver, at the nameserver, or on the way back.

Overall, our request loss rate plots give a good indication of losses in the network or at the nameservers.

Finally, we measure the total data rate in each direction on our commodity link. We compute rates every ten seconds and use this interval to build distributions for traffic rates in both directions. We were curious about traffic rates since the link in question was a rate limited (20 Mb/s) ATM OC3 circuit and we wanted to verify that the rate-limiting was not affecting our request loss rate measurements.

We collected flow data from the traffic meters at 5-minute intervals, and computed the corresponding median values. For each of the 13 root nameservers and 11 gTLD servers, we collect the number of requests sent, the response time, and the loss rate. Not all nameservers use BIND; older versions of the Microsoft servers ask only the A servers (first in the list of nameservers in a referral), resulting in disproportionately higher query rates to the A server. We discarded data intervals (of five minutes each) if there were less than 10 queries in the interval, i.e., too few queries to produce statistically defensible data for that interval.

We do see data to each of these servers due to internal algorithms in the BIND software that spread the load when there is more than one possible choice of nameserver to contact. Observing the spread of requests across the root or gTLD servers provides us with BIND's view of the state of those servers, and provides a sensitive indicator of changes in a server or its network paths.

Experimental Results

We have data from four time periods: June 2000, November 2000, January 2001 and July 2001. Results from our *commodity* meter in June 2000 showed high

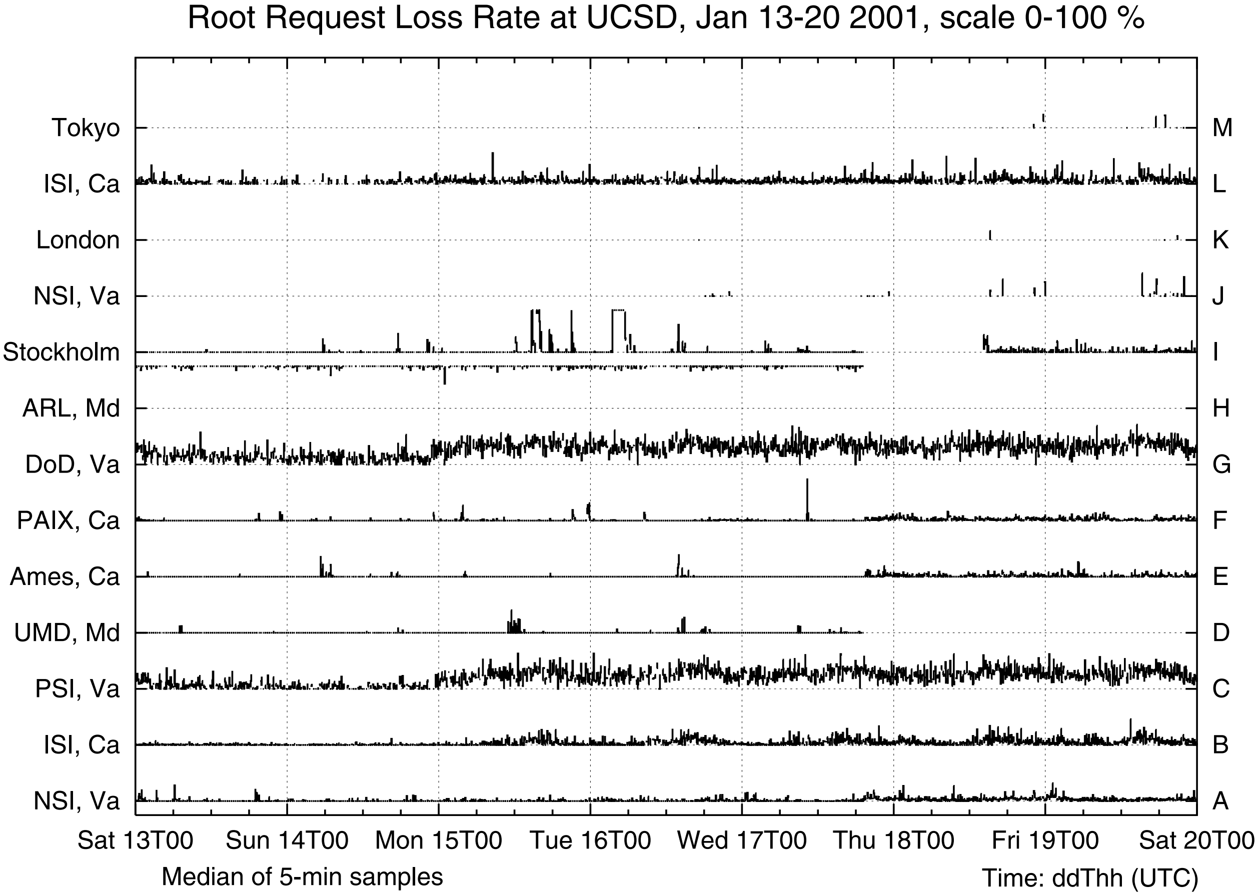

Figure 3: Heavily loaded root servers (C, G and B) show high and variable loss rates, especially during weekdays. H root has almost 100% losses until connectivity fails at 2000 on Wed 17 Jan. Other servers show short loss spikes, indicating network losses.

unanswered-request or unrequested-reply rates, primarily due to asymmetric routing. To overcome this idiosyncrasy in our topology, we chose a set of local nameservers whose requests and responses travel via the commodity link, and configured the *commodity* meter to filter out DNS packets to and from other local nameservers. This filtering reduced the 'unanswered' and 'unrequested' counts to negligible levels.

Due to local routing policy, our *commodity* meter does not see DNS request or response packets for two of the roots. In January 2001 we began using our *edu* meter, which sees traffic for all the root and gTLD servers, and higher DNS traffic rates than for the *commodity* meter. Early experience with the *edu* meter showed a significant reduction in unanswered-request rates, but it also showed 'unrequested-response' rates of the same order. This indicates that our meter was seeing asymmetric routing, possibly via the fourth (unmetered) external link.

We changed to our OC12 *edu* meter in July 2001. Data collected since then shows negligible levels of 'unrequested' responses. We believe that our request loss plots are reliable for November 2000, and from July 2001 onward. Our January 2001 data covers three consecutive weeks, for the other periods our data covers periods of only two to four days. We will concentrate on the January data to describe request rates, request loss rates and response times, and refer to the other data to see trends over time.

We use two types of graphs: strip charts and server performance '4-plots.' In the strip charts we plot data for either the 13 root nameservers or the 11 gTLD servers on a single figure with time on the x-axis and stacked narrow strips on the y-axis, each strip with data for a single server. The time period is a full week (Saturday to Friday) so workday variations, weekends and holidays are clearly visible (for example, the A and F servers in Figure 2). The server performance 4-plots show median response times, request loss rates, total query rates, and median overall load on the rate-limited commodity link.

Times on the plots are indicated in the format ddThh [14], i.e., day, T, hour of day. We use UTC times – they are eight hours ahead of the local time zone, i.e., 8 a.m. San Diego time is 1600 UTC. This, together with the Saturday to Friday format rather than the more typical Sunday to Saturday format causes the weekday peaks to appear shifted to the right.

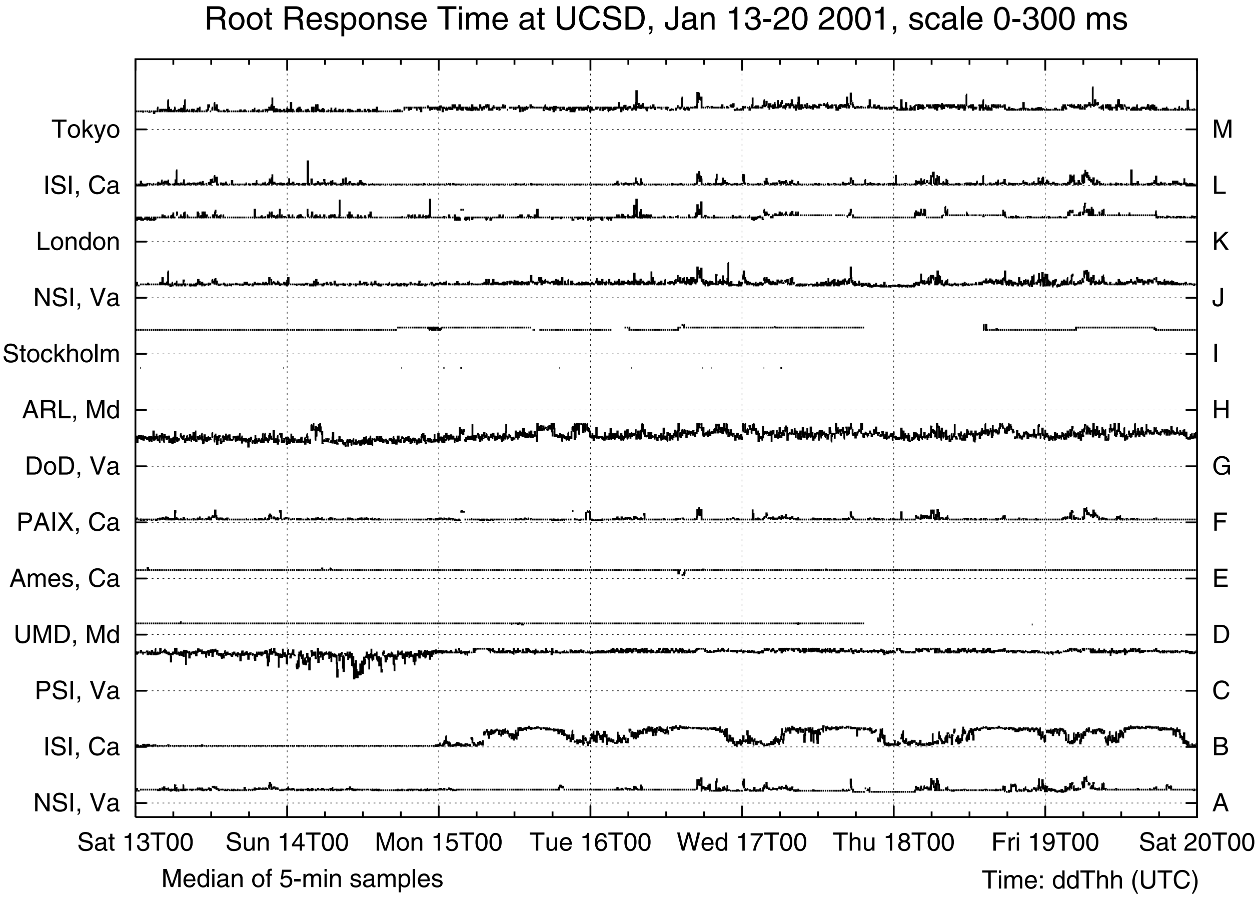

Figure 4: DNS root response time traces are normally flat with occasional spikes (A, E and F). Overloaded servers have persistently high response times, e.g., H (only occasional dots plotted), C and G. B is close to overloading, having good response times during the weekend, but poor response (with diurnal variations) during the week.

Strip Charts: Root Nameserver Performance

Our first goal in this study was to develop an understanding of how the global root and gTLD nameservers behave as viewed from our campus. We produced strip charts for the 13 root servers (A to M) and the 11 gTLD servers, plotting the number of requests, median unanswered request percentage, and median response time. Values are all plotted on the same vertical scale (shown in the title at the top of the graphs), with high values clamped at the maximum.

Figure 2 shows DNS requests to the root servers for the week January 13-20, 2001, as seen by our *edu* meter. Monday, January 15 was Martin Luther King Day, a holiday in the US; request levels are lower on this 3-day weekend than during the week. Flat tops on the curves correspond to times where the query load to that server was more than 200 per five minute interval as seen for F root during weekdays. Requests are spread across all servers, with A and F the most heavily used. BIND uses its own round trip time measurements to determine closest servers and uses those more frequently, as described in the first section.

We lost connectivity to the D, H and I roots at about 2000 on January 17; connectivity to I was restored about 1500 on January 18. Our normal routes to these servers all go via the same research network, so the plots indicate a failure in that network. The flat sections of the request plots for D, H and I (Figure 2) show that BIND reacted by dramatically reducing its request rate to those servers, waiting for them to reappear.

Figure 3 shows the percentage of unanswered requests to the root servers; we omit time intervals with too little data (<10 requests) since they are not statistically valid. Request loss rates are usually a few percent, but these losses are not generally visible to users because BIND (at the local nameserver) masks them by caching earlier responses, by retrying requests using other servers (rather than resending to the same server), and by preferring servers that respond quickly.

The H server, which had loss rates above 90% until we lost connectivity to it late on 17 January, was particularly bad. Response times for the H root are the worst on Figure 4 – they appear on the plot as occasional dots at about 360 ms.

Each server also showed request loss rate spikes in the 10% to 25% range. Four of the root servers are

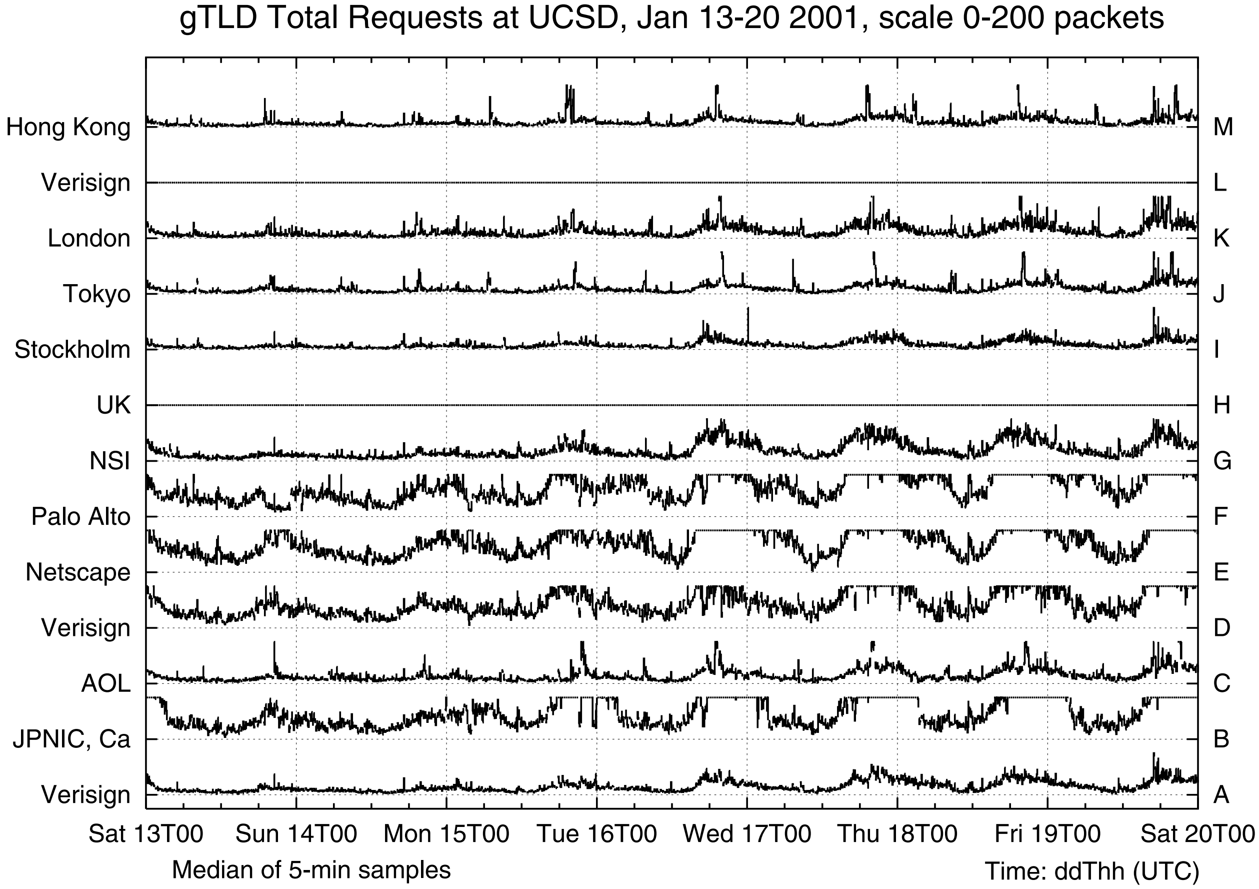

Figure 5: Our local BIND sends many more requests to gTLD servers than to roots. BIND favors those servers located on the US East Coast (B, D, E, F), producing clear diurnal variations during weekends as well as weekdays.

located in California, another six are in the Washington DC area. *traceroute* shows that packets to servers in each of these two groups follow common paths until they are close to the servers. Spikes at different times for different servers within these groups suggest congestion at points beyond the common segments of the paths. The additional delay may derive from the network near the servers, or at the servers themselves. Losses that occur across several servers at the same time indicate congestion near the measurement meter. The C and G servers show higher and quite variable loss rates, typically around 70%. Our routes to C, F and G root go via our commodity Internet link, then through different ISPs to each server. Their loss patterns and response times differ noticeably, suggesting that the observed variability for C and G indicates either congestion at or near those servers or, more likely, overloading of the server machines themselves.

Figure 4 shows observed response times for requests to reach root servers and responses to return. Roots A, F, E, K and L showed fairly flat response times, with short variations suggesting periods of congestion. Roots C and G have high response times and are highly variable, reflecting their loss behavior.

B root shows very good performance on Saturday 13 and Sunday 14 January, with response times varying between 7 and 12 ms. Over that weekend B has the lowest root response time, (the next lowest was F, 18 to 22 ms). As Figure 2 shows, BIND recognized this, and sent most of its DNS requests to B during that time. After the weekend, however, B's performance deteriorated – its unanswered request rate increased and its response time increased dramatically. BIND responded by sending requests to other root servers instead. The data is consistent with a server (B, in this case) being unable to handle its request load.

Figure 4 also shows roots A, I and K with small but marked (10 to 20 ms) extended steps in response time, lasting 12 hours or more. To determine whether these changes were due to route changes, we examined data collected by CAIDA's topology mapping project [15] during the week shown in Figure 4. Forward path topology traces were available for paths between our university and the A and K roots, collected at intervals of about 50 minutes. In that week (13-20 January 2001) we saw few forward path changes for the A root, but many forward path changes for one or two hops within the path from our

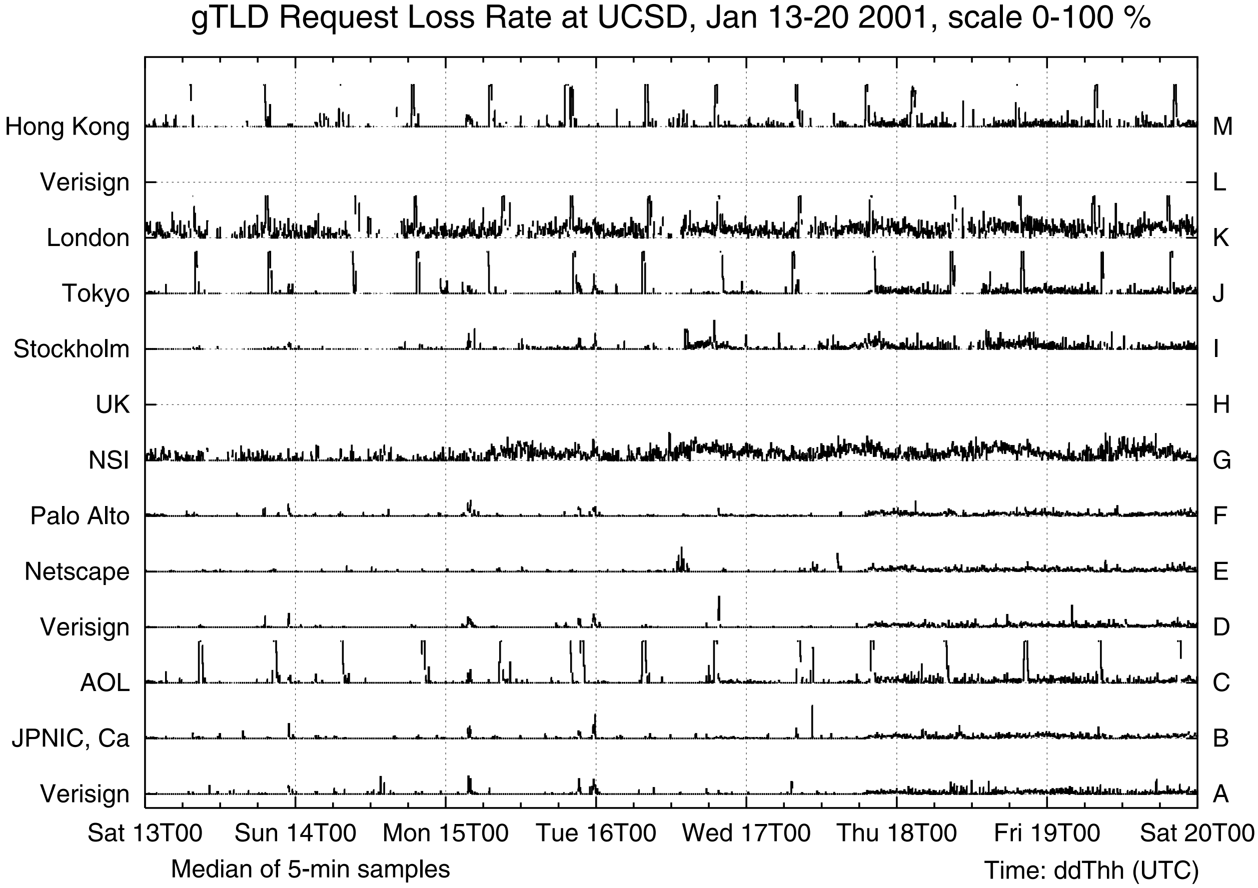

Figure 6: gTLD server loss rates show a regular pattern of spikes about every 12 hours for the C, J, K and M servers. These spikes appear when the servers are reloading their zone tables. The G and K servers show high background loss rates; since their response times remain good (Figure 7), these losses indicate network problems.

university to the K root. These path changes did not appear to be correlated with changes in round-trip time. Distribution bin sizes limit the precision of our 5-minute median response times, nonetheless they lie within one bin of the observed skitter round-trip times. We believe that the steps in our response time traces may simply be artifacts introduced by our choice of bin sizes.

Strip Charts: gTLD Nameserver Performance

Figures 5-7 exhibit generally similar behavior to the root server behavior discussed above. Most requests went to the B, D, E and F gTLDs; these four gTLDs are all located in California, so BIND favors them because they have lower response times from our campus than other gTLD servers. There were considerably more gTLD requests than root requests, reflecting the size of the domains involved.

Figure 6 is similar to Figure 3, except that it shows 5-minute median loss rates for the gTLD servers instead of the roots. We observed a regular daily pattern of 100% loss during a 30 minute period twice a day as certain gTLD servers reload the .com zone (>2 GByte). This pattern was also visible from New Zealand's meter, in spite of the substantially lower DNS request rate there.

Figure 7 shows the observed response time for gTLD servers. Response time spikes at different times for different servers suggest congestion, most likely away from the measurement point. Losses that occur across several servers at the same time usually indicate congestion near the measurement meter.

The very clear response time steps, for F at 2000 on Sat 13 Jan and for C at 1600 on Mon 15 Jan, indicate route changes in paths to or from those gTLD servers. Figure 5 shows that BIND reduces its request rate to F during the route change. We do not see a similar change in request rate for C because it is nearer to the US West Coast, and BIND was sending comparatively few requests to it. In each case normal routing and response time were restored after a few hours.

4-plots: Correlations between DNS metrics

Figure 8 compares behavior of the A and F gTLD servers from Saturday 27 to Wednesday 31 January, 2001. Routes to the A and F gTLD servers normally go via our commodity Internet link; this link's total load (i.e., all packets, not just DNS) is shown in the bottom chart of Figure 8. Both inbound and outbound traffic rates are often close to the commodity link's rate limit of 20 Mbps. The plot shows median

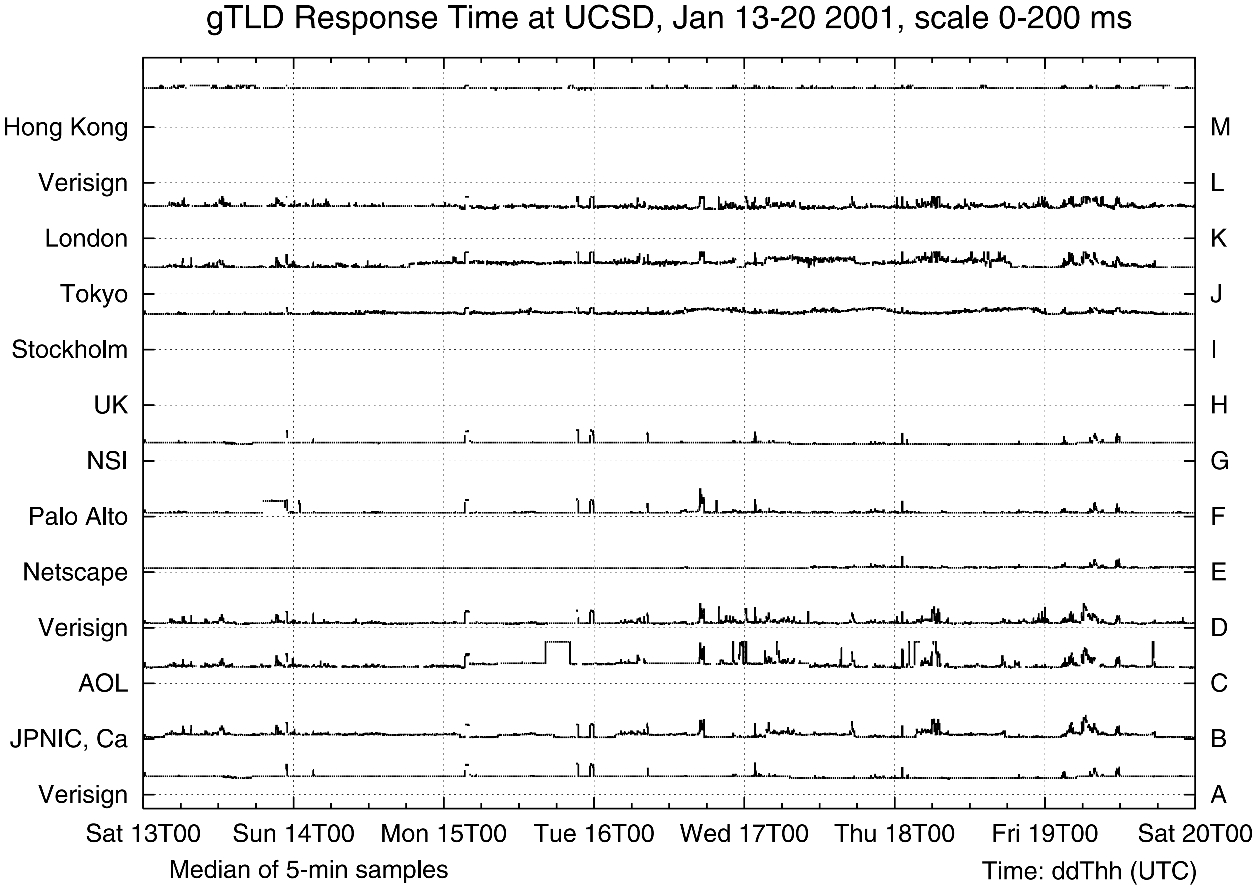

Figure 7: gTLD response times are generally steady, with short spikes common to many servers indicating network congestion. Pronounced steps, e.g., for F at 2000 on Sat 13 Jan, C at 1600 on Mon 15 Jan, indicate route changes.

traffic rate for 10-second intervals, i.e., the link was carrying more than this for half of the 10-second intervals. The link is clearly saturated (and discarding packets) for tens of seconds at a time.

Periods when the link is saturated often produce increases in response time, for example on the 28th from 0130 to 0300, the 29th from 1800 to 1900. For these two periods there is also a marked increase in the unanswered request rate, suggesting increased packet loss. However, we also see periods when the request loss rate increases with no change in the response time (e.g., from 1400 to 1600 on the 30th), or when response time increases with no change in the request loss rate (e.g., from 2300 to 2400 on the 29th). We plan to investigate correlations between these metrics in a future study.

During the four days shown in Figure 8, A gTLD's median response time drifted slowly up from about 90 ms (Saturday) to about 110 ms (Monday) and back to about 100 ms (late Tuesday). The obvious steps in its response time plot (grey line, upper subplot) are caused by our choice of binning intervals.

Overall the two servers behave similarly, showing spikes in response time at identical times. Since they are located on opposite coasts of the US, the fact that paths to both of them have problems suggests congestion close to our meter. However, they do behave differently in some ways, for example the spikes around 0300 on Sunday 28 January – which affect A but not F – and the slow drifts in A's response time. Our 4-plots provide an effective way to compare server behaviors.

Observed Behavior for Some Global Nameservers

Since we began this study in June 2000 we have reported our findings to the root/gTLD server operators, and received much useful feedback in return. The following subsections summarize some of the changes we have observed.

Filtering Bad Queries at the F Root

Figure 9 plots F root performance on Tuesday, 21 November 2000, showing a drop in response time, from 75 to 30 ms at 2230 (i.e., 1530 local time in Palo Alto). We originally thought this was due to the separation of the gTLD and root zones, but now know [16]

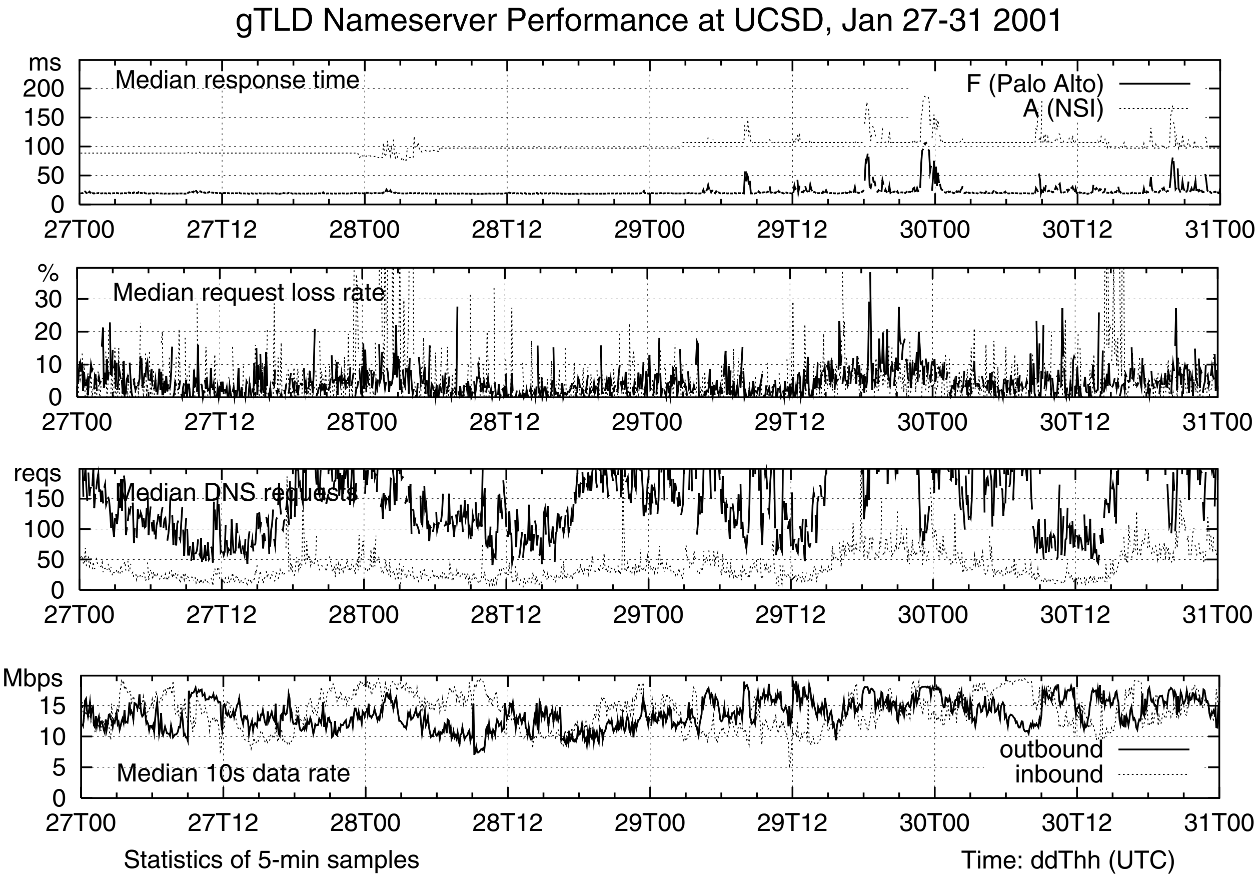

Figure 8: Performance metrics from Figures 5-7 for the F (black lines) and A (grey lines) nameservers. DNS packets for these servers travel via our commodity link. The lower plot shows commodity traffic levels; our commodity link is clearly saturated for long periods. Some of these periods also show spikes in response time or loss rate, but overall there is little correlation between traffic rate, DNS response time, request loss rate or query rate.

that the drop was due to filters introduced by the F root's operator that drop unanswerable packets. The lower response time is accompanied by an increase in the request rate due to BIND's load balancing.

Zone Reloads for the gTLDs

Figure 10 shows 5-minute median loss rates for the gTLD servers in November 2000. We observed a regular daily pattern of 100% loss during a 30 minute period twice a day while *all* the gTLD servers reload the .com zone (>2 GByte). These loss periods correspond to zone transfers, when the machine is reloading the .com zone and is too busy to answer queries.

These prominent loss spikes were also visible in our measurements from New Zealand, in spite of the low DNS request rate there.

Figure 11 shows this behavior in detail for a single day. From this view we can see that the *named.conf* file configuring BIND for these servers allows three zone transfers at a time (parameter `transfers-out = 3`). The servers reload in pairs, the sequence is A+B, C+E+J, D+G+K, and then F+M.

We showed our observations to Network Solutions, who run the gTLD servers, and they changed their reload policy [17]. Each gTLD server is a pair of machines that were both reloading simultaneously. Figure 6 shows that by January 2001, only the C, J, K, and M gTLD servers were still simultaneously reloading, and by the end of July 2001 these zone-reloading-induced losses are completely gone.

Upgrade of the H Root

Performance of the H root server in January 2001 seems abysmal. Indeed, earlier we discussed (Figure 3) the total loss of all H root's DNS requests, and we have also observed H root's poor performance from New Zealand. In January H root had a request loss rate consistently above 90%. Our request rate was low, yet the response time (shown by the occasional dots at about 365 ms on the response time chart, Figure 4) was much higher than for the A root, which is also located on the East Coast of the US. This data suggests that either the network near H or the H server itself was badly under-provisioned.

In our plots for July-August 2001 H is one of the best performing root servers. The story behind H's miraculous recovery is a new system administrator who noticed the very low query rate and load average, and insisted on a hardware upgrade.

In his words [18] (quoted with permission):
"Until about June 11th of this year the system was

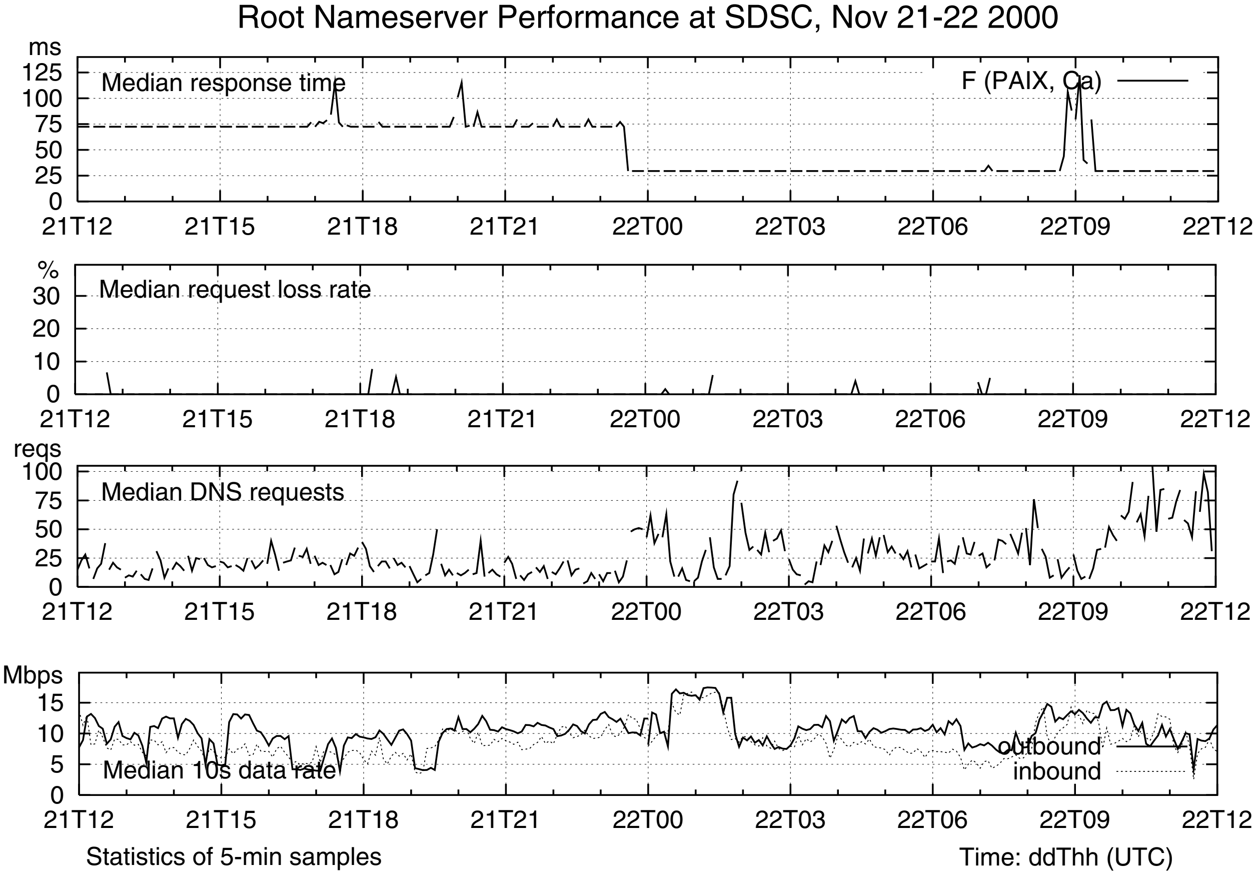

Figure 9: Filtering for unanswerable root queries was introduced on F root at about 2230, Tue 21 Nov. Server response time dropped, and our local BIND responded by increasing its request rate to F.

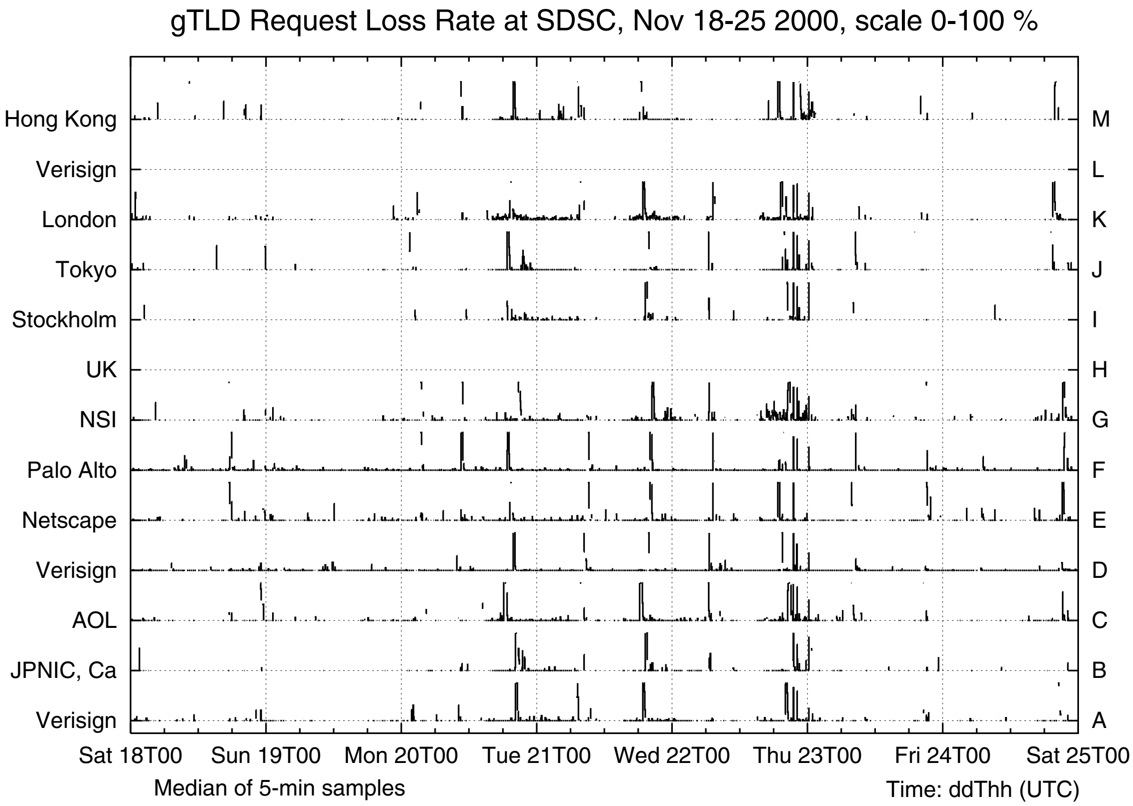

Figure 10: gTLD request loss rate in November 2000, showing high loss (i.e., periods when several servers were not responding) during zone reloads for *all* servers.

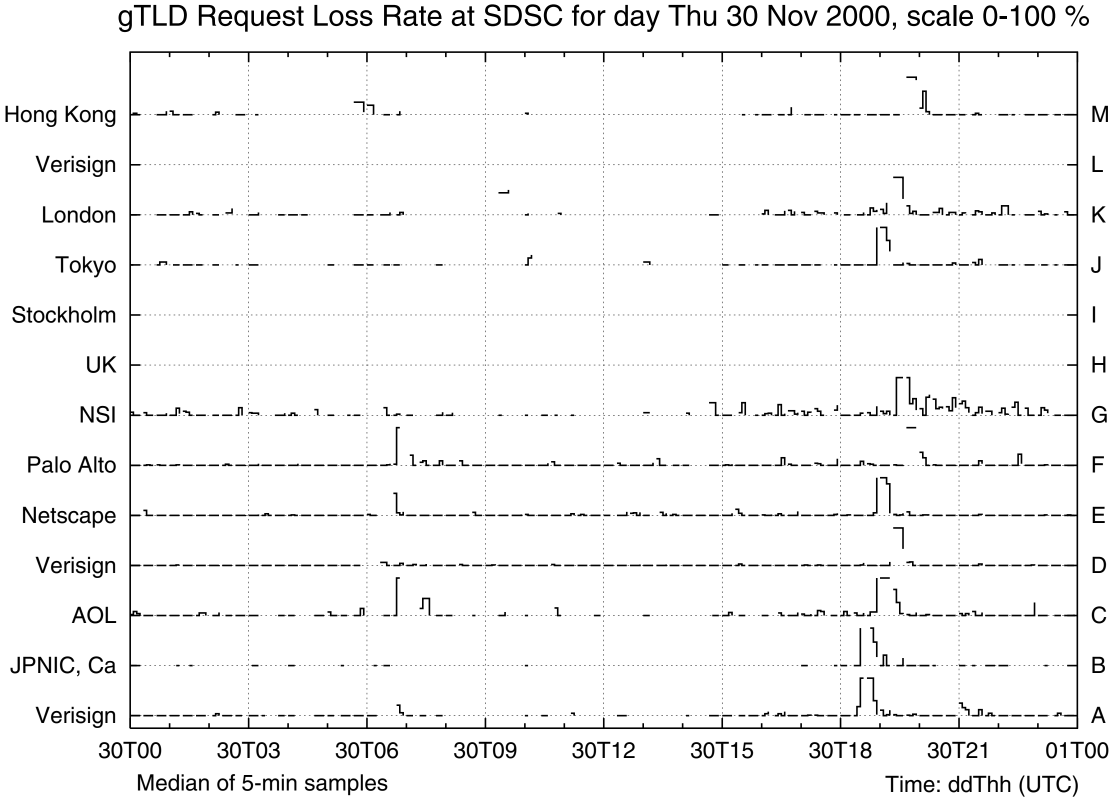

Figure 11: Details of zone reload loss pattern on Thu 30 Nov, showing periods when three servers at a time were inaccessible.

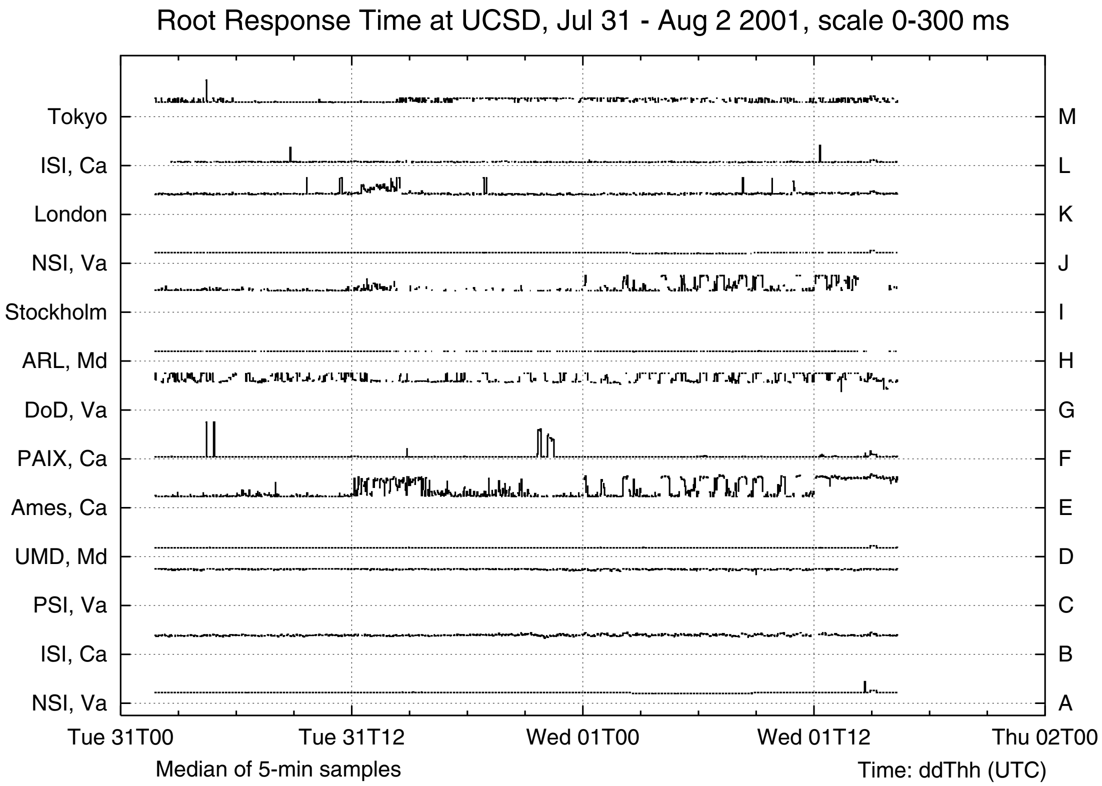

Figure 12: Root server response time in late July 2001. H response is vastly improved, A, B, J and L responses are less variable compared to their response time in January. E and I response times, however, have become more variable.

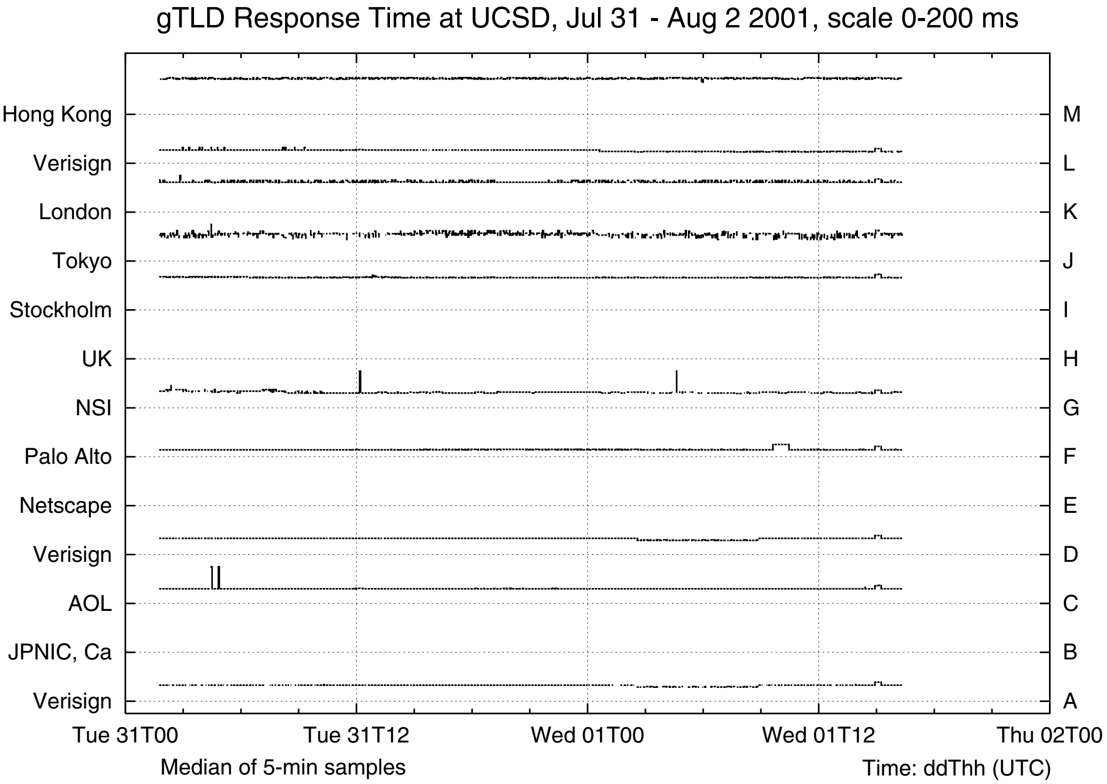

Figure 13: gTLD server response time in late July 2001. J response times remain unchanged, but all the other servers show less variable response times.

running on a 168 MHz Sun Ultra-2. I contend that this, alone, was the reason for the poor performance. After taking over 'H' from the previous administrator, I noticed that 'H' was only able to receive and respond to about 1800 queries/second even though thousands of more queries were being sent to it. Around June 11th, the system was replaced with a >1.2 GHz Intel system and is now 'seeing' between 4000 and 7000 queries per second on average and responding to every one. I've even seen peaks of up to 11,000 queries per second with an equal response rate. I've been collecting statistics since the new server came online if you are interested (10-minute averages). Based on these, the maximum 10-minute average query rate has been 7332.43 queries/second with an average 10-minute query rate of 4259.48 queries/second (8079 data points).

As for H's connectivity, we have a OC12 connection to the Defense Research and Engineering Network (DREN): http://www.hpcmo.hpc.mil/Htdocs/DREN/index.html .

This network has numerous peering locations with the global Internet at such locations as SprintNAP, MAE-East, MAE-West, Fix-West, Seattle GigPoP, Chicago NAP, and transit service through AT&T WorldNet. All are OC3 connections."

Long-term Observations on the Roots and gTLDs

Figure 12 shows 5-minute median response times to the root servers in late July 2001. Our *commodity* meter could not see DNS traffic to the B, E, H or I roots, hence they only appear in our 2001 plots. We have data for the other root severs for 2000 and 2001.

H root now has steady, low response times; this is the most dramatic performance improvement we have observed. Otherwise, the long-term average of the root servers 5-minute median response times have not changed significantly since we began measuring them. We also observe a decrease in the variability of response times, especially for the B root, which was highly variable in 2000, but is steady in 2001. The A, J and L roots had variable response times through January 2001, but these are now (i.e., in July 2001) quite stable. On the other hand, some root servers have become more variable during 2001, for example E and I.

Measured request loss rates have decreased overall. Some of this decrease is due to improvements in our experimental technique, in particular the deployment of our *edu* meter, which greatly reduces the likelihood of missing data due to asymmetric routes from the meter to/from the root and gTLD servers. Another contributing factor is the gTLD servers, which became became operational during the third quarter of 2000, reducing the query load on the root servers, and the traffic congestion on paths to them. In addition, there have also been improvements both in the root servers Internet connectivity and in their capacity to cope with offered load of DNS queries.

Figure 13 shows the gTLD servers 5-minute median response times in July 2001. The J server's response time remains rather variable, i.e., it has not changed since November 2000. However, the degree of variability in response time for the other gTLDs has decreased since November 2000, to the point where their response times are now steady. In November 2000 and January 2001 we saw spikes in response time that were common to all gTLDs located in the US. This effect is no longer visible in July 2001.

In summary, the root and gTLD operators have made considerable efforts to improve the performance of their servers, resulting in worthwhile and observable improvements in the stability of both root and gTLD servers over the last year.

The gTLD servers are well-provisioned, using a single hardware and software architecture. They are well-run, by a single operator. The roots are also well-run, by different operators, using different (sometimes older) hardware and software architectures. This diversity makes them less prone to failures common to a single architecture; we believe that in the long term this hardware and operating system diversity is a desirable feature.

Conclusions and Future Work

Our passive traffic meters have provided an effective method to monitor performance of global nameservers as seen from a client site perspective. Once a meter is configured, 5-minute distributions of response times, together with counts of requests and request loss rates, provide ample data for monitoring performance of the global nameservers and our Internet links to them.

The 'total requests' charts (e.g., Figures 2 and 5) reflect local user activity. They also show which roots and gTLDs are most used by local nameservers; sudden changes indicate a loss of connectivity to one or more global nameservers.

The request loss rate charts (e.g., Figures 3 and 6) show which servers are experiencing short-term congestion or connectivity/routing problems. Problems affecting Internet links close to the measurement site show correlated changes for many of the global nameservers, while problems affecting more distant links appear only on the charts for single servers.

The 'response time' charts (e.g., Figures 4 and 7) indicate how long it takes the root/gTLD nameservers to resolve a DNS query. This metric is important since the delay is often directly visible to, and detrimental to performance for, end users.

The efficiency of the local caching component of the DNS architecture, together with system administrators setting DNS TTLs of the order of minutes to days (rather than seconds to minutes), greatly improves user-perceived delay, since fewer requests need global lookups. Overall, BIND's load balancing and caching on local nameservers, makes the global DNS extremely reliable (by impressive design).

The DNS 'server performance' plots provide some insight into the way response time and request rate vary with the load on our commodity Internet link. The request loss rate did not, however, show any obvious correlation to other metrics. More work is needed to understand these relationships.

These client-side measurements show that most root servers, as observed from our university, have reasonably low response times, but only fair request loss rates, perhaps due to the rate-limiting on our commodity Internet link. Some servers (C and H for example, and to a lesser degree G) had consistently high loss or high latency or both. Hardware performance may be a contributing factor. Considerable performance improvements have occurred this year, but further investigation of the client-side root/gTLD server measurements is necessary to enable a longer term view of systemic infrastructural performance issues.

Our strip charts are useful for day-to-day monitoring of our Internet connectivity, since they reveal changes in network behavior on paths between our local network and the global servers without our having to send test packets. We are now developing a monitoring tool that will produce these charts in near real time.

A NeTraMet meter, using either live network data, or *tcpdump* trace files, is a useful network administration tool in several ways, including:

- **Monitoring remote DNS servers**, as detailed in this paper. Bear in mind that as well as monitoring the global servers, our strip charts provide a clear indication that a site's local nameservers are load balancing properly.
- **Monitoring local servers**, for example Web servers. As well as measuring request loads and server response times, one might monitor the load balancing between several servers attached to a common network segment.
- **Monitoring traffic on external links**. This traditional use for NeTraMet, i.e., traffic flow measurement, is useful for detecting link saturation, capacity planning, accounting and billing, etc.

NeTraMet is distributed as open-source (GNU Public License) software. It may be downloaded, together with its documentation, from the NeTraMet web site [3]. An introduction to NeTraMet, with discussions on how to configure and use it, is given in [6].

Acknowledgements

We gratefully acknowledge the generous help of our colleagues in implementing the CoralReef version of NeTraMet, in setting up our commodity meter and (at very short notice) our *edu* meter, and in producing diagrams for this paper. Thanks to Hans Werner Braun (NLANR), Bud Hale (NLANR), Bradley Huffaker (CAIDA), Ken Keys (CAIDA), Joerg Micheel (The University of Waikato), David Moore (CAIDA) and Brendan White (CAIDA).

Author Information

Nevil Brownlee co-chaired the IETF's Realtime Traffic Flow Measurement (RTFM) Working Group. He created NeTraMet, an open-source implementation of the RTFM (Internet Standard) architecture at The University of Auckland late in 1992. Nevil now works half time in Auckland overseeing technology developments, particularly those relating to networks, and teaching in Computer Science. He spends the other half of his time at CAIDA in San Diego, where he pursues research into the behavior of Internet traffic, and continues to develop NeTraMet so that it can handle higher-speed networks. Nevil can be reached at nevil@caida.org .

kc claffy is principal investigator for CAIDA, the Cooperative Association for Internet Data Analysis, based at the University of California's San Diego Supercomputer Centre. kc's research interests include: data collection, analysis, and visualization of Internet workload, performance, topology, and routing behavior. She also works on engineering and traffic analysis requirements of the commercial Internet community, often requiring ISP cooperation in the face of commercialization/competition. kc can be reached at kc@caida.org .

Evi Nemeth has been a computer science faculty member at the University of Colorado for years teaching data structures, networking and system administration. In 1998 she visited CAIDA at the University of California, San Diego, where she led the IEC (Internet Engineering Curriculum) effort, and continues to contribute to various CAIDA research activities. Evi is a co-author of the UNIX System Administration Handbook, now in its third edition. She is now moving out of the UNIX and networking worlds and onto her 40 foot sailboat to start exploring the real world. Evi can be reached at evi@caida.org .

References

[1] P. Mockapetris, *Domain Names – Concepts and Facilities*, Internet Standard 0013 (RFCs 1034, 1035), November, 1987.

[2] BIND website, http://www.isc.org/products/BIND/ .

[3] NeTraMet website, http://www.auckland.ac.nz/net/NeTraMet/ .

[4] CoralReef website, http://www.caida.org/tools/measurement/coralreef .

[5] C. Huitema and S. Weerhandi, *Internet Measurements: the Rising Tide and the DNS Snag*, Monterey ITC Workshop, September, 2000.

[6] N. Brownlee, *Using NeTraMet for Production Traffic Measurement*, Intelligent Management Conference (IM2001), May, 2001.

[7] N. Brownlee and M. Murray, *Streams, Flows and Torrents*, PAM2001 Workshop, April, 2001.

[8] N. Brownlee, kc. claffy, M. Murray and E. Nemeth, *Methodology for Passive Analysis of a University Internet Link*, PAM2001 Workshop, April, 2001.

[9] N. Brownlee, C. Mills and G. Ruth, *Traffic Flow Measurement: Architecture*, RFC 2722, October, 1999.

[10] N. Brownlee, *SRL: A Language for Describing Traffic Flows and Specifying Actions for Flow Groups*, RFC 2723, October, 1999.

[11] S. Handelman, S. Stibler, G. Ruth, *RTFM: New Attributes for Traffic Flow Measurement*, RFC 2724, October, 1999.

[12] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot and E. Lear, *Address Allocation for Private Internets*, RFC 1918, February, 1996.

[13] N. Brownlee, kc. claffy and E. Nemeth, *DNS Measurements at a Root Server*, Globecom 2001, November, 2001.

[14] M. Kuhn, *A Summary of the International Standard Date and Time Notation*, http://www.cl.cam.ac.uk/mgk25/iso-time.html .

[15] CAIDA Topology Mapping website, http://www.caida.org/tools/measurement/skitter/ .

[16] Private Communication, Paul Vixie, September, 2001.

[17] Discussion with NSI staff, Wed 13 Dec, 2000.

[18] Private Communication, Howard Kash, August 2001.