Response time distributions for global name servers

Nevil Brownlee, Ilze Ziedins

Abstract—
Our earlier measurements of global name server performance concentrated on response time measurements. In this paper we examine the shape of response time distributions. These distributions often show clear evidence of multipathing behaviour. We also report on improvements to NeTraMet's method of collecting data distributions.

I. Introduction

Since late 2000 we have been making passive measurements, observing the behaviour of the global root and gTLD DNS servers. We use a NeTraMet meter located at UC San Diego, referred to as our *UCSD* meter [5]. This meter measures the time interval between DNS requests and their corresponding responses [11], producing either individual times, or distributions (with counts in up to 100 bins), depending on the number of observations made in each 5- or 10-minute interval.

Previous work [3], has examined the long-term behaviour of the global DNS servers, using strip charts to show variations in median request/response time and in the number of unanswered requests for each 5- or 10-minute interval. This paper

- presents improvements to NeTraMet's collection method
- investigates the effect of multipathing on DNS response times and
- reports preliminary investigations of the shape of the request/response time distributions.

II. Data Collection

The data used in this paper was collected from July 2001 onwards. We collected DNS response time (RTT) data using NeTraMet [4] [6] meters, with rulesets to observe streams of DNS packets to and from all the global root and gTLD servers.

A. Network Topology, Meter Location

For this paper, our data was obtained from our *UCSD* meter, as described in [5]. During the year the UCSD network topology changed several times, changing our ability to meter external Internet traffic. The data used for this paper was collected in July and September 2001, and does not appear to have been affected. However, the data used to investigate DNS resolver retry behaviour was collected in January 2002. It reveals clear changes (discussed below) in the routing of DNS packets past our meter.

Figure 1 shows the network topology at the end of 2001. San Diego Supercomputer Centre (SDSC, right of figure) has four external links, one to the commodity Internet and three to research and higher education networks. The existing OC3 ATM link from SDSC to the rest of the University (UCSD, left of figure) was replaced about mid-year by an OC12 ATM link. Our *UCSD* meter uses Dag 3.2 network interfaces [1] which work at OC3 or OC12 speeds, so we moved the meter to observe the OC12 UCSD link.

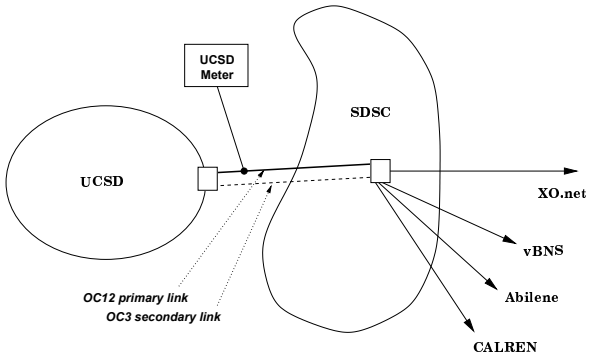

Fig. 1. Network topology at UCSD, December 2001

Later in the year, an OC3 ATM link was added between SDSC and UCSD. That OC3 link is available as a secondary connection, providing redundancy for the primary OC12 link. The secondary link is now in everyday use, which means that our *UCSD* meter can no longer reliably see all packets in and out of UCSD. For DNS packets, local routing determines the paths to each of the global name servers; we analyse the implications for our data below. In the long term we plan to install a second meter on the OC3 link so as to restore our ability to observe all packets in and out of UCSD.

B. RTFM Distributions in NeTraMet

The 'basic' RTFM attributes [7] all have scalar values which are either static (e.g. `FromPeerAddress`), or are integer counters (e.g. `ToOctets`). RTFM counters are never reset; instead one reads an RTFM meter at regular intervals and computes differences between the counts. Using counters in this way allows a meter to be read asynchronously by several meter readers.

RFC 2724 [9] extended the RTFM data model by introducing *distribution-valued* attributes, allowing an RTFM meter to produce data about how an attribute's value varies over time. The RTFM Working Group decided that distribution values were more general (and therefore more useful) than simple statistics such as mean, median, etc. Furthermore, since a distribution is simply an array of counters, it can be read asynchronously by multiple meter readers.

The essential parameters of an RTFM distribution are shown in figure 2. A distribution is an array of bins (n bins in the figure). An attribute's values, in the range $(lower\_limit\ ..\ upper\_limit]$, are mapped onto the bins using either a linear or logarithmic transform. Bin 0 counts all values $<= lower\_limit$, and there is an $n+1^{th}$ 'overflow' bin which counts all values $> upper\_limit$.

Within the RTFM architecture, each rule in a ruleset can test

Nevil Brownlee is with The University of Auckland, New Zealand and CAIDA, SDSC, UC San Diego, E-mail: nevil@caida.org

Ilze Ziedins is with The University of Auckland, New Zealand, E-mail: i.ziedins@auckland.ac.nz

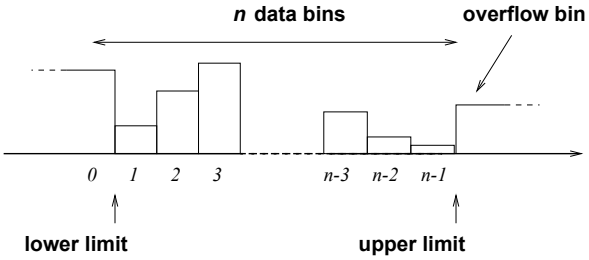

Fig. 2. RTFM Distribution parameters. Note that the limits specify the upper edge of each bin.

a value under a mask. The 'value' and 'mask' fields in a rule are at least six bytes long (so as to hold an Ethernet MAC address). RFC 2724 specifies how the complete set of distribution parameters is coded as fields within an RTFM rule as follows:

```
Mask bytes:
    1   Transform       1 = linear, 2 = logarithmic
    2   Scale Factor    Power of 10 multiplier for
                           Limits and Counts
  3-4   Lower Limit     Highest value for first bin
  5-6   Upper Limit     Highest value for last bin

Value bytes:
    1   Bins            Number of bins.  Does not
                           include 'overflow' bin
    2   Parameter-1     } Parameter use depends
  3-4   Parameter-2     } on distribution-valued
  5-6   Parameter-3     } attribute
```

When writing rulesets using SRL [10] one requests the meter to build a distribution using a `save` statement of the form

`save d_v_attribute = value & mask;`

and specifying the parameters using SRL's conventions to indicate field size in bytes, i.e.

- a dot indicates that the preceding number is a one-byte integer,
- an exclamation mark indicates that the preceding number is a two-byte integer, and
- the last field is the same width as its preceding field.

For example, we could specify 'short-term bit rate' [9] distributions as follows:

```
save ToBitRate   = 48.10.0!0 & 1.3.1!24000;
save FromBitRate = 48.10.0!0 & 1.3.1!24000;
# 48 bins, 10s rates, linear, **3 => 1k..24M B/s
```

`ToBitRate` and `FromBitRate` are measured in bits per second, the multiplier of 3 converts them to kb/s. These attributes use `parameter-1` to specify the interval for computing rates, 10 seconds in the `save` statements above.

C. DNS Resolver Behaviour at UCSD

The NeTraMet ruleset we use for observing DNS RTT distributions uses the `ToTurnaroundTime1` attribute. The ruleset defines one *flow* for each global root or gTLD nameserver. Each of those flows has a *stream* [11] for each local DNS resolver; these streams maintain a queue of data blocks for DNS request packets. When a DNS request packet is observed, its arrival time (in microseconds) and its DNS identifier are saved in the appropriate stream's packet data queue. When a DNS receive packet arrives, NeTraMet searches its stream's packet data queue looking for a matching request identifier. If a matching request is found, the meter computes the RTT as the difference in arrival time for the response and request packets.

From time to time the meter checks the packet queues for 'old' requests; these are timed out (i.e. deleted) if they have been queued for more than ten times the $upper\_limit$ specified for the RTT distribution. We normally use an RTT range of 1 .. 700 ms, hence requests are timed out after 7 seconds. Timed-out requests are counted in the flow's `ToLostPackets` attribute.

Sometimes a meter may see response packets when it did not see a corresonding request. In that case, the unrequested response is counted in the flow's `FromLostPackets` attribute. Unrequested responses indicate that the meter is not seeing all traffic in both directions for a link – in contrast, timed-out requests can be caused either by a failure to see both directions or by losses in the network.

Since we know (figure 1) that we are only metering the primary link between SDSC and UCSD, we expect to observe path asymmetries for some of the global nameservers. To determine the effect of this on our RTT data, we modified the NeTraMet meter to log copies of its DNS packet data. On 16 January 2002 we collected about seven hours of DNS requests and responses, with one data record for each DNS (UDP) packet. Each record contains

```
Local DNS resolver address    = SourcePeerAddress
DNS request ID (2 bytes)
Arrival Time (microseconds)
Global server address         = DestPeerAddress
DNS parameter (2 bytes)
```

For those seven hours we observed 111 local resolvers active on the UCSD campus. The number of successful lookups (i.e. request/response packet pairs) is shown in table I.

TABLE I
DNS LOOKUPS (I.E. REQUEST/RESPONSE PACKET PAIRS) BY NUMBER OF RETRIES AT UCSD ON 16 JANUARY 2002

DNS Lookups		
No retries	98432	(98.3%)
One retry	1584	(1.6%)
Two retries	91	(0.1%)

We observed similar behaviour for unanswered requests, but such counts are unreliable because we cannot be sure whether or not a response was delivered via another path (in our case, UCSDs OC3 secondary link). Nonetheless, none of our local resolvers attempted more than *two* retries. Recently Jung et al [2] have observed DNS lookups involving up to 12 retransmissions. Their results, however, are for all nameservers queried by their local resolvers, whereas ours (showing a maximum of two retries) are only for the global root servers.

In the following examples the records are set out with their fields in the order listed above. Times are shown in seconds and ms, DNS identifier and parameter are shown as four-digit hexadecimal numbers. A request packet has a zero high-order bit in its parameter field.

Normal behaviour for a resolver is to send a request and receive a response from the same global server, e.g.

```
LocRes1  0001  003879.297  C gTLD  0000
```

```
LocRes1  0001  003879.373  C gTLD  8000

LocRes1  0004  008977.691  M gTLD  0000
LocRes1  0004  008977.900  M gTLD  8000
```

Here local resolver `LocRes1` sent requests to the C and M gTLD servers, and recived responses within a few hundred milliseconds.

Resolver retry behaviour depends on the resolver implementation. A typical example is:

```
LocRes2  44fd  017923.609  G root  0000
LocRes2  44fd  017927.125  I root  0000
LocRes2  44fd  017931.126  B root  0000
LocRes2  44fd  017931.132  B root  8000
```

In this case `LocRes2` sent a request to the G root. After about 3.5 seconds it sent the same request (i.e. a request with the same identifier) to the I root. 4 seconds later it retried to the B root, and got a response. We describe this as 'normal' retry behaviour.

Occasionally we saw non-standard retry behaviour, in which a resolver sent duplicate packets to the same nameserver, e.g.

```
LocRes3  0364  003795.650  J gTLD  0000
LocRes3  0364  003795.650  J gTLD  0000
LocRes3  0364  003795.772  J gTLD  8000

LocRes3  6850  007116.567  C root  0000
LocRes3  6850  007116.567  C root  0000
LocRes3  6850  007120.330  D root  0000
LocRes3  6850  007120.330  D root  0000
LocRes3  6850  007120.401  D root  8000
```

Here `LocRes3` sent two copies of its requests to the J, C and D roots, and received a reply for each request. This behaviour is clearly implementation-dependent, we only observed it for a few of UCSD's local resolvers. We have also observed a few even more bizzare retry behaviours.

Duplicated requests will disrupt NeTraMet's response-matching algorithm. Because the second request is placed at the head of the packet data queue, it will be matched by the first response, giving a shorter than expected request/response time. During the observation period covered by table I we observed 260 duplicate request/response pairs, i.e. 0.3% of the total lookups. We do not believe that this percentage is high enough to have any significant effect on our RTT data.

D. Asymmetric Routing of DNS packets at UCSD

We have also used the DNS packet data described above to investigate the paths of request and response packets by building tables of requests and responses for each local resolver, and for each global nameserver.

The local resolvers all seem to behave in much the same way, i.e. we did not find any unusual DNS traffic patterns amongst them. We therefore summarised the request/response counts for each global nameserver. We find that in January 2002 our *UCSD* meter never saw requests to F root, it only sees responses. Similarly, it saw many requests to G gTLD, but very few responses. Both of these are clear examples of asymmetric routing.

Such asymmetric routing has a definite impact for our work on global nameserver performance. In particular, we cannot be sure of the 'request loss' rate to any global server, and we are unable to measure RTTs for some severs. However, the secondary link only appears to have been carrying traffic from the beginning of 2002; we will install a meter on the secondary link as soon as possible.

For this paper, our data was collected around between July and September 2001. The median RTT strip charts presented below are generally similar to those in our earlier work [3], i.e. they were not affected by changes in the UCSD network topology.

E. 'Dynamic' Distributions in NeTraMet

Using a set of bins with fixed upper bounds, as described above, works well most of the time, but it presents difficulties if one wants to observe small variations within a single flow, e.g. for a single nameserver. This is because the root nameserver response times cover a wide range, 15 ms to about 200 ms or more. On the one hand, we want to use the same scale for all the nameservers (to simplify comparisons), on the other hand we often find that all the counts for a nameserver are clumped into a very small proportion of the distribution's bins.

One way around this problem would be to use different bounds for each of the servers. That would provide better resolution, but it would make our rulesets more complex, and thus more difficult to maintain.

To avoid having to set a wide bin range, thereby giving away the fine detail we want for each flow, we have devised a 'dynamic' distribution management scheme for NeTraMet. This stores individual data values as they arrive, until we run out of space to store them. At that point we compute suitable bounds, derived from the observed data, and initialise the resulting fixed-bin distribution with the data values.

In an SRL ruleset, one specifies the scale factor and bounds as usual. $lower\_limit$ and $upper\_limit$ are held as $lowerlim$ and $upperlim$, together with R, the specified range, i.e. $upperlim - lowerlim$.

We begin by saving each data value in the space allocated for the distribution's bins, and updating min_val and max_val (the attribute's max and min data values). By default the meter has space for at most 100 bins, so we can store up to 100 data values.

When we reach the 101^{st} data value, we copy the values into a temporary array, determine suitable distribution bounds, and place the data values into their appropriate bins. After that, each new data value goes into the appropriate bin, exactly as if we had specified fixed bounds in our ruleset. We use a linear transform for dynamic distributions. Since the bounds are chosen automatically there is very little need for a logarithmic transform.

When we come to choose bounds for a dynamic distribution, we set them to $min(min\_val - R/8, lowerlim)$ and $max\_val + R/4$; i.e. we pick values which allow some room for 'outlier' points at the ends of the range we have seen so far. This usually works well, but pathological cases do arise. We have not yet been able to find a more effective strategy.

When a dynamic distribution is read by a NeTraMet meter reader, it returns a value of the 'transform' parameter to indicate whether it is a set of actual values or an array of counters (together with the chosen upper and lower bounds). The NeTraMet 'transform' parameter values are:

```
1  linear          } as above
2  logarithmic     }   (RFC 2724)
```

```
4  dynamic request     Use in SRL program

5  actual values       } Returned by
6  array of counters   }   meter reader
     (limits set from data values)
```

After the distribution values are read by our meter reader, the meter resets them to zero and begins to build the next dynamic distribution, as above. Since resetting distribution values in this way breaks the RTFM model, where all data is assumed to be held in counters which only ever increment, it can only be used with a single meter reader. We are considering ways of making this approach work with multiple meter readers, but for our current project a single meter reader is sufficient.

F. Strip Charts of Root and gTLD Response Time

The data for this paper was collected using a NeTraMet rule set based on the one described in [11]. Because we wished to determine whether there are differences in the behaviour of the various IP networks (i.e. blocks of IP addresses) inside UCSD, we modified the ruleset by adding a statement to save each flow's SourcePeerAddress, which in this context is the IP network address from which DNS requests are sent.

Since our flow data files can include several different flows for each global server (one for each different SourcePeerAddresses) we wrote a perl program to combine the distributions from such flows. The result is a 'combined' flow data file, with a single flow for each global server. Figures 3 and 4 show 'strip charts' of overall DNS round-trip time (i.e. median request-response times) for 10-minute intervals during the eight days we collected our data.

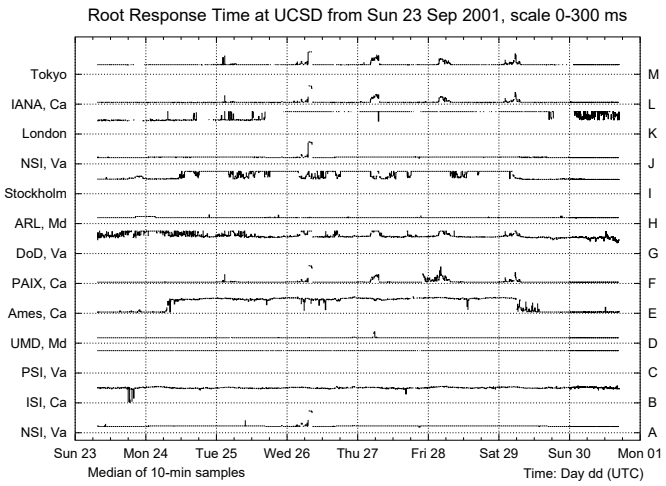

Fig. 3. DNS root performance: response times observed at UCSD for eight days from Sunday, 23 September 2001

Examining the root traces, three features stand out. First, seven servers were performing well (A, D, F, H, J, L and M), i.e. they had low response times which were mostly steady. Second, several servers show various types of overloading behaviour. B, C and G showed consistently high response times, with no difference between weekdays and weekends. E, I and K had periods of a few days (mostly weekends) when performance was reasonable, but much higher response times (usually during the week). Third, several servers, especially F, G, L and M had short periods when response time was noticeably higher than usual. Since these periods coincide across several servers, they are most likely caused by network congestion, rather than overloading at individual servers.

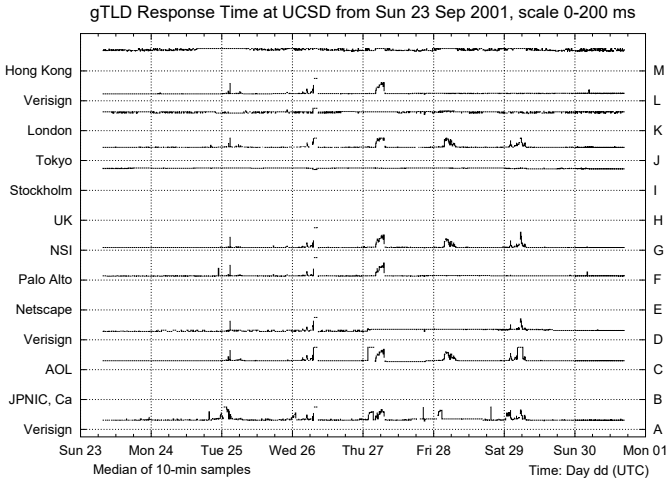

Fig. 4. DNS gTLD server performance: response times observed at UCSD for eight days from Sunday, 23 September 2001

The gTLD traces (figure 4) are more consistent than the roots. They do not show any obvious server overloading effects, but they do show short-term increases in round-trip time, at the same times as those observed for the roots. This behaviour is most likely caused by network congestion in a common part of the network paths for these global servers.

III. Fitting Distributions

Some preliminary data collection and analysis of DNS request response times had been done for an earlier set of data using bins with fixed upper bounds. Gamma, lognormal and Weibull distributions were fitted, but no single distribution was always succcessful, although it seemed that the best choice of distribution might depend on the server. For instance, the Weibull gave the largest number of good fits for A root, whereas the lognormal appeared to fit some of the F gTLD data better. This preliminary fitting of distributions was complicated both by the small number of responses observed in some five-minute periods; and the narrow range of some of the observed distributions compared with the fixed bounds chosen for the bins, which led to high counts in just a few bins. The dynamic distribution management scheme has overcome both of these difficulties.

A particularly interesting feature of the dynamic distribution scheme is that it stores response times in the order in which they were collected, as long as there are no more than 100 of them. Thus it is possible to generate time series plots of DNS response times for those 10 minute collection intervals where the number of responses is below 100. Note that the observations for these plots are taken at *unequal intervals*.

We begin by giving some representative plots of some of the behaviours that were commonly observed. The right-hand column of figure 5 contains time series plots of request response times for L root from subnetwork 1 for a sequence of 10 minute

intervals, beginning with the 10 minute interval 3:40-3:50 a.m. on Thursday 27th September, and ending an hour later at 4:40 a.m. The left-hand column of the same figure gives the fitted density using the function *density* in R with default parameter settings. The statistical package R [8] has been used to do the analysis and generate all of the plots in the remaining sections of the paper.

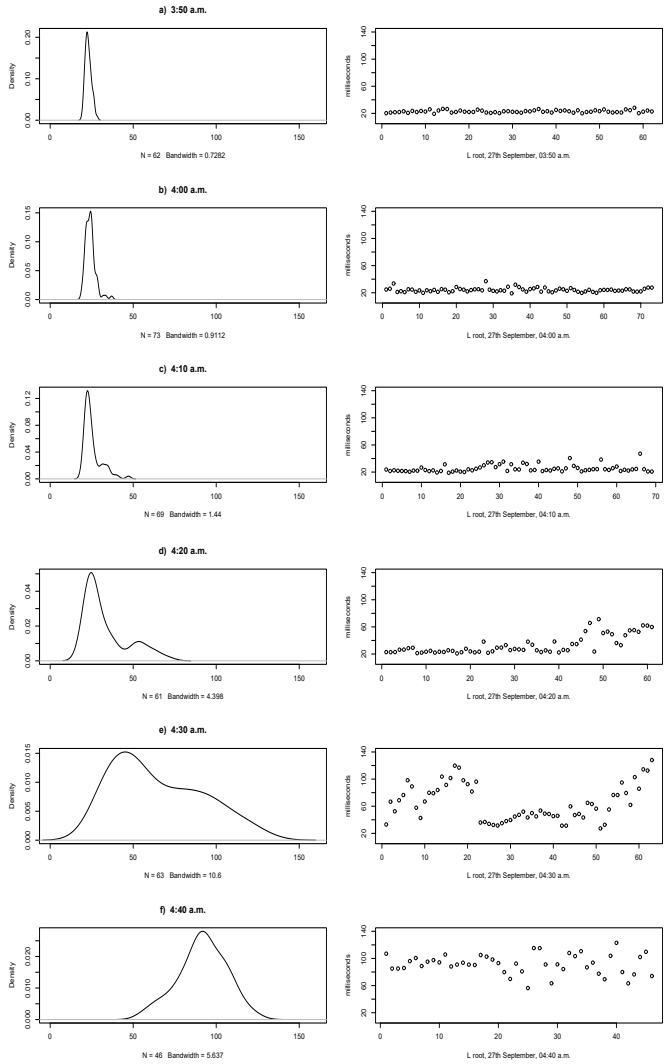

Fig. 5. L Root, sample ending 0440, 27 September 01 (UTC) showing increase in DNS request response times over the period of an hour, with fitted density using R

If we examine the plot for the first 10-minute interval, from 3:40-3:50 a.m., we see that there is relatively little variation about the mean. The mean is 22.95 milliseconds, the sample standard deviation 1.85, and the skewness 0.54. Figure 6 gives qqplots for the normal, Weibull, gamma and lognormal distributions (from left to right). Both the Weibull and gamma distributions give a reasonable fit in this case. The gamma distribution has estimated shape and scale parameters 2.71 and 18.64 respectively, with location parameter 19.4. This behaviour is typical of lightly loaded routes, with a gamma distribution with a positive location parameter often giving a reasonable fit.

The sequence of plots for the whole hour clearly show that the response times increased considerably over this period. Indeed, for the final 10-minute interval, the data have mean 91.67 milliseconds, sample standard deviation 14.59 with skewness -0.27. The qqplots (which we do not give here) indicate that a better fit is obtained with the normal distribution than the gamma for this interval. The increase in request response times over this period was also seen by D, F, G and M root servers (see Figure 3), i.e. it happens during a period of network congestion common to those servers.

In some cases, the increase in request response times is to a level that is many times the base level. Figure 7 gives the distributions for a sequence of 10 minute intervals on E root on 24 September (again, this increase is clearly visible on the strip chart in Figure 3). In figure 7 a) the median is around 16

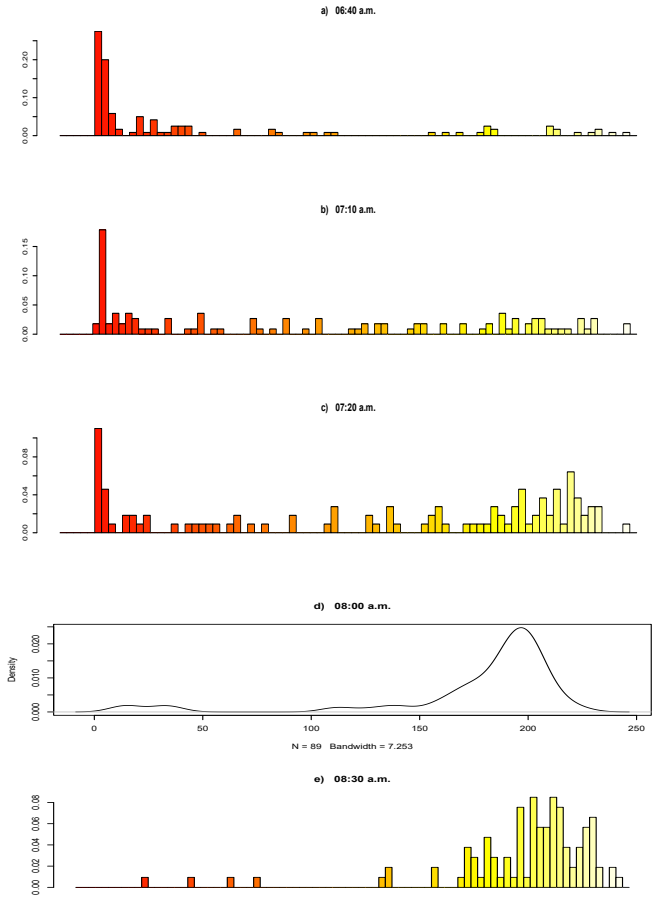

Fig. 7. E Root, samples ending 0830, 24 September 01 (UTC) showing an order of magnitude increase in response times.

milliseconds, with some excursions above this, and the distribution is right skew. Figures 7 b) - e) show the response time distributions gradually increasing to a level where the median is about 200 milliseconds, that is, an order of magnitude higher, at which it remains for the rest of the day. In these plots most of the 10 minute intervals have had more than 100 response times recorded – the data are binned, and the plot is now a barplot instead of a time series plot. However, note that it is possible to visually compare the fitted density for the time series data with that of the barplots.

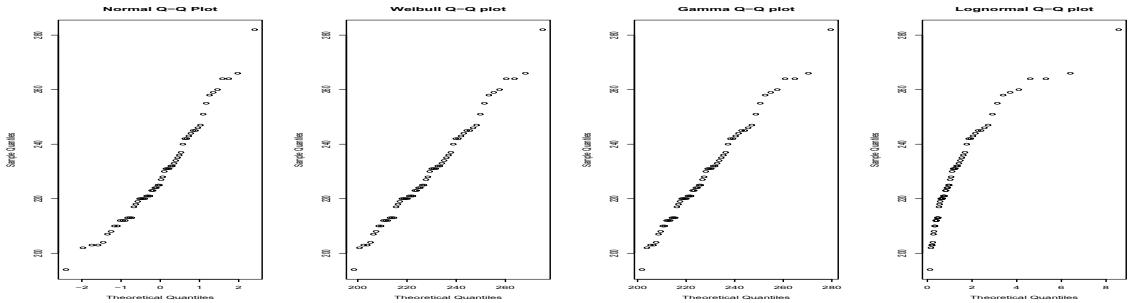

Fig. 6. L Root, sample ending 0350, 27 September 01 (UTC), Q-Qplots for Normal, Weibull, Gamma and Lognormal distributions (reading from left to right). Gamma and Weibull give the best fit.

If we compare the distributions for Monday, 24 September with response time distributions on Sunday, 23 September (see e.g. those given in figure 8), we see that on Sunday they mostly remain at the lower level for the whole day. Note that pings on E root take about 30 milliseconds, on average – the high levels of response times seen for much of the working week are an order of magnitude greater than this.

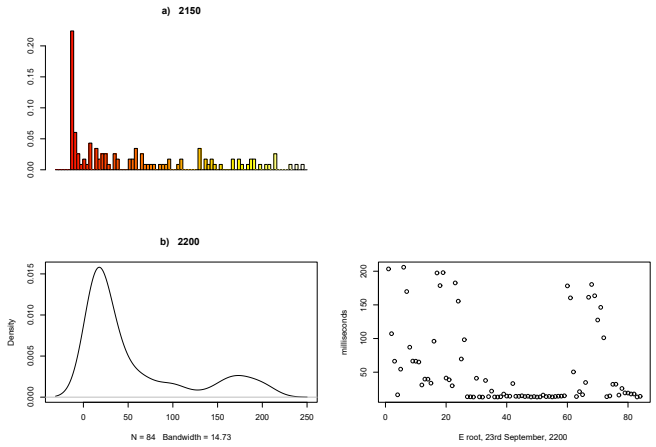

Fig. 8. E Root, sample ending 2200, 23 September 01 (UTC) showing typical behaviour for a day with light loading.

The plot in figure 8 b) of response times is a behaviour that is very commonly seen. Fairly long periods of low response times are interrupted by short clusters of longer response times. The tail of the distribution depends heavily on how many of these clusters are present and how long they are. We found that if the higher-valued clusters are omitted, then the gamma distribution often gave a reasonable fit to the data. In general, however, no single distribution appears to give a consistently good fit to the data.

A. Multimodal behaviour

A particularly striking feature of the data is that we see clear evidence of multipathing, even over very short time spans. We have observed this effect in data collected over several weeks.

We observed two general kinds of multipathing behaviour:

(a) Several clear modes, with round trip times differing by about 10 milliseconds.

(b) A shift from one mode to another, with round trip times differing by hundreds of milliseconds.

One reasonable explanation for the first behaviour is the use of load balancing by network operators to spread traffic across several paths. Our observations suggest that this practice is surprisingly widespread. The second suggests either a significant change in routing along the path or a change in server behaviour – overloading of the server seems the more likely explanation.

A plot showing request response times in July 2001 for the whole UCSD network is given in Figure 9. This is an example

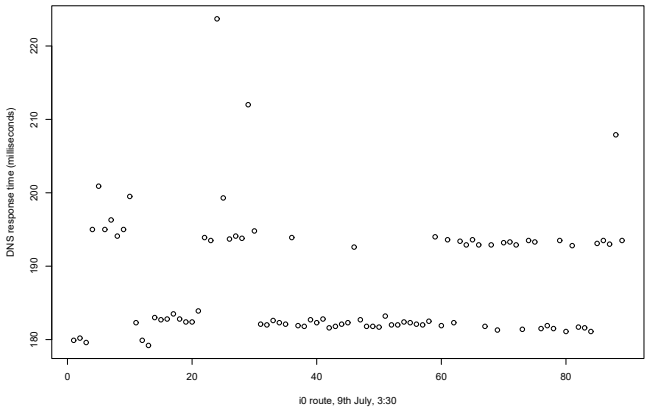

Fig. 9. I Root, sample ending 0330, 9 July 01 (UTC) showing preferred values of the response times: we believe that these are due to multipathing.

of the first kind of behaviour. We observe that there are preferred values of the response times: we believe that these are due to multipathing. It is evident from figure 9 that multipathing can produce multiple route changes (which manifest as different DNS response times) over relatively short periods of time. We have seen this preferred-value behaviour occurring for several of the root servers and gTLDs. K and M gTLDs exhibited changing levels in response times in most of the 10 minute intervals for which we had data; I and L gTLDs in many of the 10 minute intervals; and occasionally rapid switching was also seen for A, B and E root servers, and A, C, F and G gTLDs. We note that some routes had so little data available that it would not have been possible to detect this behaviour if it were present.

In those cases where multipathing was clearly present, we fitted distributions separately to the different response time modes.

Again, a gamma distribution was most often found to give a reasonable fit, with the Weibull also giving a good fit on many occasions. However, the frequent presence of very different response time modes means that fitting a distribution is non-trivial, and it would be more difficult to implement this as an automatic computation.

B. Subnetworks

In our September data collection, DNS request times were also classified according to the subnetwork from which they originated. We found that the distribution of request times for some of the gTLDs could vary between different subnetworks. An example of this is given in figure 10 below, where response times from subnetwork 1 are all above 85.9 msecs (with one outlier of 354.9 msecs which has not been plotted), whereas those from subnetwork 2 all lie in the relatively narrow range of 84.2 to 85.4 milliseconds.

Since the traffic from all our metered subnetworks goes through the SDSC router, differences for subnetwork round trip times should not be caused by routing differences. This suggests that they are caused by load balancing at the servers.

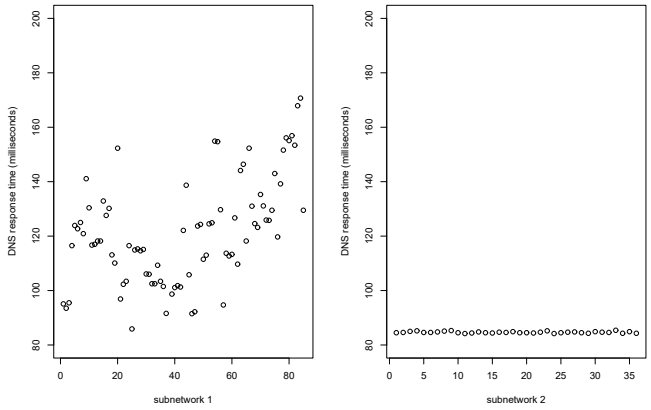

Fig. 10. F gTLD, subnetworks 1 and 2, sample ending 0450, 26 September 01 (UTC)

The different response time modes observed for the whole network were also visible for individual subnetworks. Fig 11 gives a 10 minute interval from K gTLD subnetwork 1. K gTLD displayed such behaviour consistently and repeatedly over long periods of time. This appears to be type a) multipathing, as in figure 9.

C. Common Paths

On several occasions it was possible to observe very similar behaviour for several different root servers and/or gTLDs at the same time.

For instance, on 26th September, in the 10-minute interval ending at 03:10, an increase in the response times was noted for F, J and L roots and C, D, F and G gTLDs. Figure 12 gives the time series plots of response times for some of these routes. It must be remembered that these are time series plots – there is no record of when in the interval the response times were gathered, just the sequence in which they were gathered. Thus, although

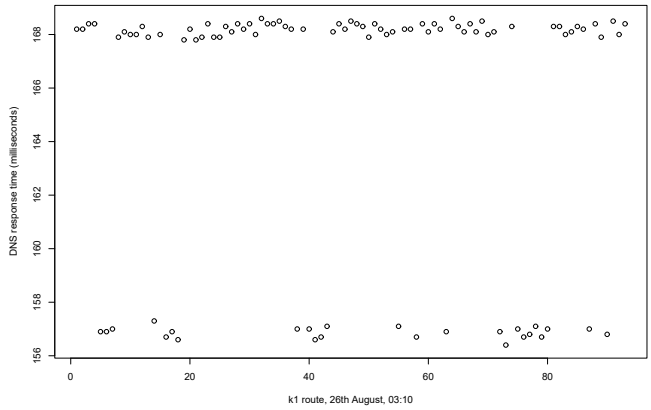

Fig. 11. K gTLD subnetwork 1, sample ending 0310, 26 September 01 (UTC)

the increases appear to occur simultaneously, it is possible that they occurred at very different times within the interval.

On occasion, it appears that multipathing may be occurring simultaneously on several routes. An example of this is in the 10-minute interval ending at 0700 on 26 September, for A, C and possibly also M gTLDs – see figure 13. Note that the order of magnitude of the change appears to be the same for all plots, which suggests that the paths to these three servers share a common mulipathed section.

IV. Summary

We describe an improved method of collecting data, which entails storing the actual data values obtained during a collection interval, as long as there are no more than 100 observations. If there are more than 100 observations, they are stored in bins, the bounds for which are calculated dynamically using the first 100 observations as a guide. Resolver retry behaviour, and its effects on NeTraMet's DNS request response packet matching is discussed.

Our improved method of collecting distribution data provides greatly improved fine detail for our DNS response data. We have observed clear evidence of multipathing in around 50% of servers and we are surprised at how common this is. Multipathing means that fitting a single common distribution will not be possible, and automatic fitting would be nontrivial. We found that a gamma distribution with a location parameter often gives a good fit, at least for identified modes, with the Weibull also giving a good fit on many occasions.

For monitoring purposes, median and interquartile range for the response times might be sufficient, possibly with some measure of skewness as well; using these summary measures could reduce network data collection overheads. However, we would then require a new algorithm so that a meter could compute percentiles for 'the last n minutes of data.'

Acknowledgements: The authors are grateful to the network support team at CAIDA and to our colleagues in the WAND group. This research was partly funded by a FRST-NERF grant UOWX0011.

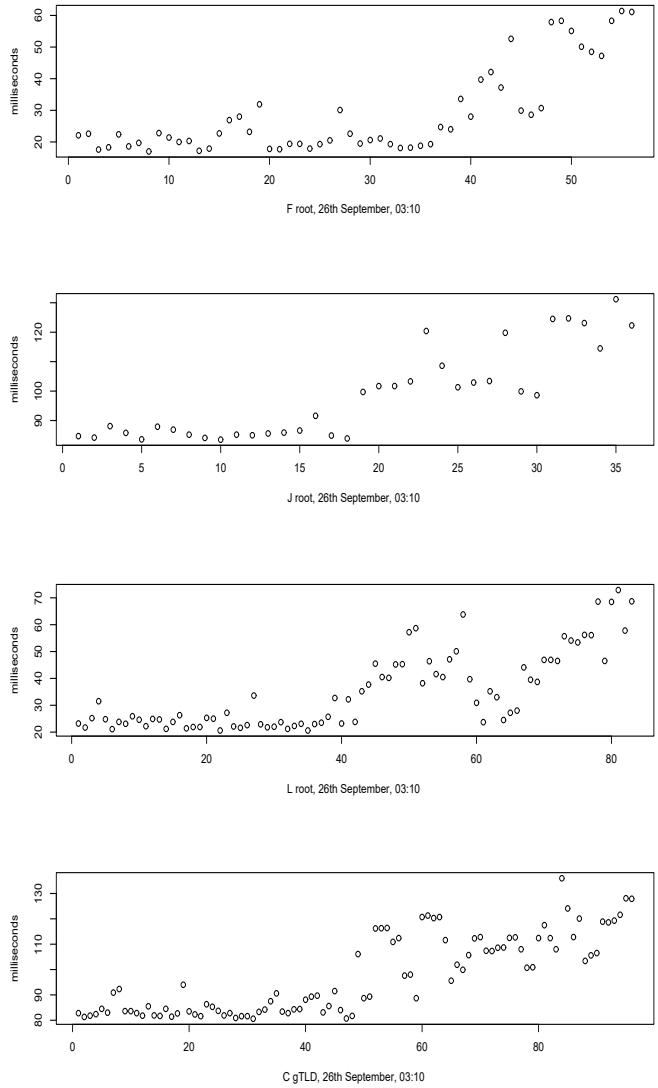

Fig. 12. sample ending 0310, 26 September 01 (UTC) showing possibly simultaneous increase in response times on F, J, L roots and C gTLD.

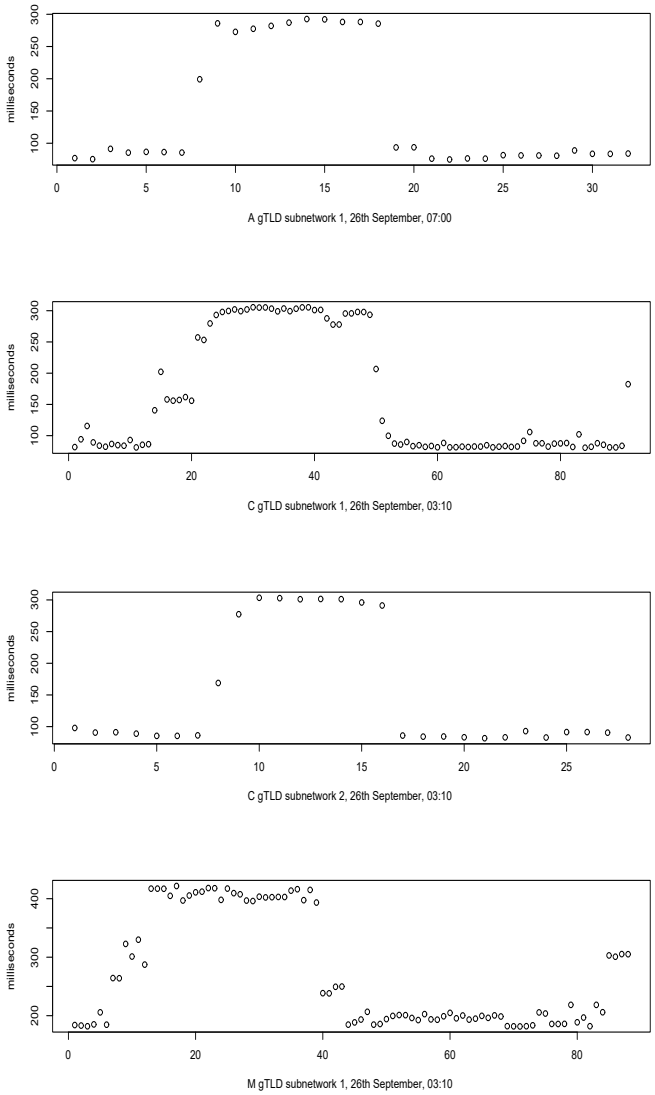

Fig. 13. sample ending 0700, 26 September 01 (UTC) showing possibly simultaneous multipathing on A, C subnetwork 1, C subnetwork 2 and M gTLDs (from top to bottom). Note that here subnetworks 1 and 2 both seem to experience simultaneous multipathing to C gTLD.

REFERENCES

[1] Dag cards website, http://www.endace.com/products/products.html

[2] Jaeyeon Jung, Emil Sit, Hari Balakrishnan and Robert Morris, *DNS Performance and the Effectiveness of Caching*, ACM SIGCOMM Internet Measurement Workshop, November 2001, available at http://www.icir.org/vern/sigcomm-imeas-2001.program.html

[3] N. Brownlee, kc claffy and E. Nemeth, *DNS Root/gTLD Performance Measurement*, Usenix LISA Conference, December 2001

[4] Nevil Brownlee, *Using NeTraMet for Production Traffic Measurement*, Intelligent Managament Conference (IM2001), May 2001

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

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

[7] Nevil Brownlee, Cyndi Mills and Greg Ruth, *Traffic Flow Measurement: Architecture*, RFC 2722, Oct 1999

[8] R Project for Statistical Computing, http://www.r-project.org/

[9] Sig Handelman, Stephen Stibler, Nevil Brownlee and Greg Ruth, *New Attributes for Traffic Flow Measurement*, RFC 2724, Oct 1999

[10] Nevil Brownlee, *SRL: A Language for Describing Traffic Flows and Specifying Actions for Flow Groups*, RFC 2723, Oct 99

[11] N. Brownlee and M. Murray, *Streams, Flows and Torrents*, PAM2001 workshop paper, April 2001