NETWORK TOMOGRAPHY, DELAY ESTIMATION & BOTTLENECK LINK DISCOVERY

Nick Johnson[*]*, John Thompson & Steve McLaughlin*

Institute for Digital Communications
School of Engineering and Electronics
The University of Edinburgh
Edinburgh, EH9 3JL
Nick.Johnson@ed.ac.uk

Francisco J. Garcia

Agilent Laboratories, Scotland
South Queensferry
Edinburgh
EH30 9TG
Frankie_garcia@agilent.com

ABSTRACT

An important issue in the measurement of networks is the ability to infer characteristics of internal network links from measurements made on end-to-end paths. It may be impractical in terms of equipment, time or cost to monitor each individual link but it is often feasible to monitor a number of existing paths. Provided there is enough traffic flowing through enough different paths then it is possible to estimate some characteristics of each link. In this paper we compare two methods for estimating the end-to-end delay distributions, one based on the method-of-moments and the other on a Gaussian approximation. This information can then be used to compute packet delay on any link in a network and then detect which link has the highest latency. This procedure is often termed bottleneck link discovery.

Index Terms— Network Tomography, Delay Distribution Estimation, Network Inference, Estimator Comparison

1. INTRODUCTION

The term network tomography is first used by Vardi in [1]. In [2], [3] & [4] the term network tomography is used to define an approach to infering network characteristics from a limited subset of measurements made in a wired network. We consider here a specific type of network tomogrpahy, the problem of estimating link-level characteristics from path-level measurements. This approach is used in [5] [6] [7] [8] because it is often impractical and inefficient to measure all internal links in a network. It is possible to infer from a set of measurements taken over a selected set of paths (where a path is a combination of links) the likely delay on each link. From these estimates a method of detecting the link with the highest delay can be used to detect a bottleneck link. Once detected, the link can be modified to reduce the delay or the network routing can be changed to lower the volume of traffic on that link.

There are two methods of gathering an estimate of the delay mentioned in the above papers. One method (used in [2]) attempts to estimate the Cumulant Generating Function (CGF) of the delay using measurements of the delay of probe packets. Another (used in [4]) is to assume an a-priori delay distribution then estimate the parameters of this distribution from the delay of probe packets. Our contribution is to use the ns2 simulator [9] to compare the two methods' ability to correctly identify the bottleneck link. Performance with a reduced number of probe packets and the computational complexity of both methods will also be studied.

The remainder of this paper is organised as follows. In Section 2 we review the methods used in [2] and [4]. In Section 3 we present results from simulations comparing both methods and an estimate of computational efficiency. Finally, in Section 4 we give some conclusions and provide pointers to further work.

2. SYSTEM MODEL & ESTIMATORS

2.1. GENERAL MODEL

A network of routers can be modelled as a set of connected links with the connections specified in a routing matrix. We define a path to be a connected set of two or more links from the total set of links L. An estimate of the distribution of delays is formed from the delays of unicast probe packets on various paths whose total number is P (See Fig 1, originally from [3] where $P = 5$ & $L = 4$ for illustration). The vector of delay observations on path i is represented by $Y_i, i = 1...P$ with the routing matrix represented by H which is of size $P \times L$. The objective is to find an estimate of the distribution of delays on each link, represented by $X_j, j = 1...L$. Equation 1 shows the linear relationship between these three quantities:

$$Y = HX \tag{1}$$

[*]Funding for this work is provided by Agilent Technologies via a PhD Fellowship Award

For the network shown in Fig 1 we define H as:

$$H = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (2)$$

Provided H is full rank then it is possible to use a least-squares (LS) algorithm to recover X:

$$X = H^{-1}Y \quad (3)$$

We define H^{-1} as the pseudo-inverse of H as in [3]:

$$H^{-1} = (H^T H)^{-1} H^T \quad (4)$$

This pseudo-inversion must be performed each time the topology changes to ensure the correct weights in the LS algorithm. For a wired scenario, as considered in this paper, is it likely to be infrequent operation.

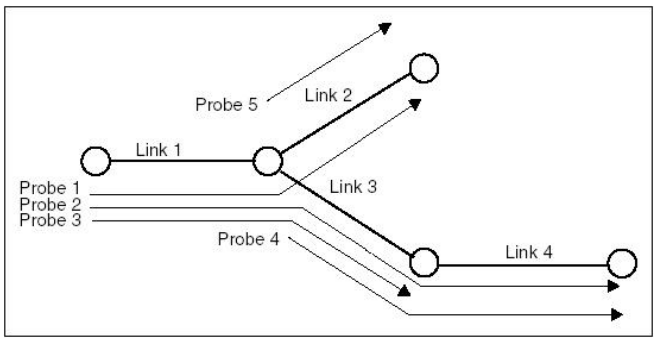

Fig. 1. Network Topology & Probe Paths

Once we have an estimate for the distribution of delay (or other network parameter) on a link it is desirable to compute the bottleneck link.Where we have delay distributions, we examine the cumulative distribution function (CDF) of all links and compare them. We seek to find the link that has the largest value which, when they are normalised, entails finding the CDF with the heaviest tail. To do this we supply a value, normally δ, which is the value at which to compare the CDFs. We choose the link with heaviest tail as our first choice bottleneck link and could continue, if necessary, to select the second, third etc. choice links. It should be noted that bottleneck link detection is not always performed and not always useful. Consider the case of a network where each link has a similar performance but with small perturbations. A bottleneck link detection method would likely identify one link as poorly performing where its performance is comparable to others.

In the remainder of this section, we will introduce the techniques under consideration.

2.2. Method of Moments

In [3] the authors estimate the CGF of the distribution of delays on each path from individual delay measurements with a method-of-moments (MoM) estimator, yielding Y. These are passed through the LS algorithm to give a CGF of the delay distributions for each link in the network.

We first construct an estimate of the CGF of path i using N measured delays denoted $Y_{ik}, k = 1...N$,

$$\widehat{M_{Y_i}(t)} = \frac{1}{N}\sum_{k=1}^{N} e^{tY_{ik}} \quad (5)$$

Then we use LS to obtain a link-level estimate of the CGF where h_{ij} is the i^{th} row and j^{th} column element of H^{-1} and hence we sum the weighted contribution from each path towards that link.

$$\widehat{K_{X_j}} = \sum_{i=1}^{P} h_{ij} \times \log(\widehat{M_{Y_i}}) \quad (6)$$

To find the link with highest delay a Chernoff upper bound is imposed on the link CGFs. The link with the highest probability, P_j of exceeding the delay threshold, δ is taken to be the bottleneck link in the network.
In [3] this is expressed as:

$$P_j = P(X_j \geq \delta) \leq e^{-t\delta} E[e^{tX_j}] \quad (7)$$

This method necessitates a-priori selection of the value of δ to be used as the delay threshold.

2.3. Gaussian Approximation

In [4] the authors suggest the CDF of delays on links could be modelled as a single Gaussian distribution.

We can estimate the mean of the delay distribution on path i as:

$$\widehat{M_{Y_i}} = \frac{1}{N}\sum_{k=1}^{N} Y_{ik} \quad (8)$$

And similarly the variance:

$$\widehat{\sigma^2_{Y_i}} = \frac{1}{N-1}\sum_{k=1}^{N} (Y_{ik} - \widehat{M_{Y_i}})^2 \quad (9)$$

We express the distribution of delays on a particular link, X_j, as a single Gaussian by using LS thus:

$$X_j = \mathcal{N}(\sum_{i=1}^{P} \widehat{M_{Y_i}} \times h_{ij}, \sum_{i=1}^{P} \widehat{\sigma^2_{Y_i}} \times |h_{ij}|^2) = \mathcal{N}(\widehat{M_{X_j}}, \widehat{\sigma^2_{X_j}}) \quad (10)$$

Note that we model the variance as a noise process so multiply by $|h_{ij}|^2$ in order to preserve the positive sign.

To find the bottleneck link we evaluate the erfc function (which normally applies to $\mathcal{N}(0,1)$ and is modified in Eqn 11) at δ for each X_j and select the link which has the highest value of P_j. Again, this necessitates a-priori selection of δ.

$$P_j = \text{erfc}(\frac{\delta - \widehat{M_{X_j}}}{\widehat{\sigma^2_{X_j}}}) \tag{11}$$

3. SIMULATION STUDY

3.1. Simulation Setup

To test both methods we use an ns2 simulation to model a wired network with unicast probe-path traffic. The topology is as shown in Fig 1 and the simulation parameters are identical for both methods.

Background traffic on each link is formed by combining a number of exponentially distributed UDP and a number of TCP traffic sources in a similar manner to that used in [2]. On each link we add a delay to each packet to force a situation where one link has higher latency than the others for both methods to detect. Aside from the added delay, other delays encountered by packets come from self-congestion due to background traffic in the form of queueing and processing time at each node. Key simulation parameters are given in Table 1.

Parameter	Value
Added Delay Link 1	100 + [10-60] ms
Added Delay Link 2	100 ms
Added Delay Link 3	80 ms
Added Delay Link 4	10 ms
Bandwidth on each link	1 Mb
Simulation Time	1000 s
Number of Paths, P	5
Number of Link, L	4
CGF Parameter, t	20
Value of δ for comparison	0.15
Number of samples, N	3000
Estimator Rate	2 Kb/s
Estimator packet size	40 Bytes
Background Traffic Link 1	800 kb UDP, 1 TCP
Background Traffic Link 2	600 kb UDP, 1 TCP
Background Traffic Link 3	900 kb UDP, 1 TCP
Background Traffic Link 4	300 kb UDP, 3 TCP

Table 1. Key Simulation Parameters

3.2. Key Results

In this section we present results showing the comparison between both estimators (the method-of-moments (MoM) and the Gaussian approxiamtion (MoG)) for different observation window sizes and estimator rates.

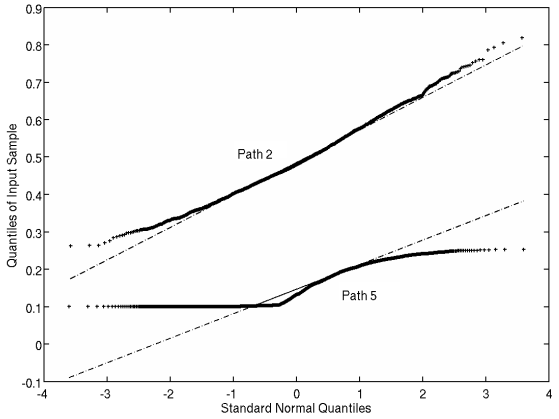

Fig. 2. QQ plot of data on paths 2 & 5

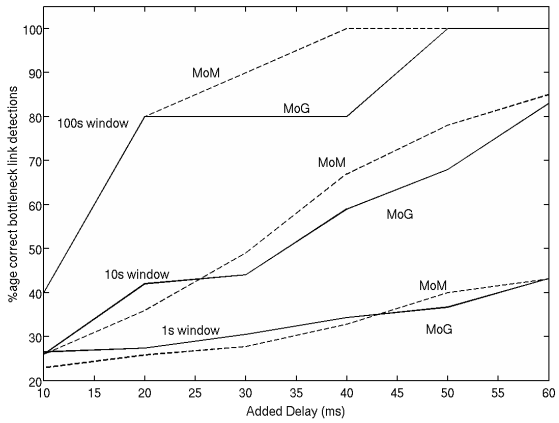

Fig. 3. Added Delay against Correct Detection Rate

To compare both methods we perform the simulation then compute the bottleneck-link for 3 scenarios:

- 1000 realizations of a 1s observation window
- 100 realizations of a 10s observation window
- 10 realizations of a 100s observation window

In Fig 2 we see a quantile-quantile plot of the delays of estimator packets on paths 2 & 5. The fit between the solid line (the data) and the dashed line (a Gaussian) indicates how well a single Gaussian models the data. On path 2, a single Gaussian distribution gives a good model of the data whereas on path 5 a single Gaussian gives a poor fit. The poor fit in the lower tail of path 5 is evident on other links and is a result of packets having a minimum delay which causes deviation from the single Gaussian model. As the fit for most paths is good in the central 4 quartiles we use the single Gaussian distribution as an estimator.

In Fig 3 we see that with the longest observation window (100s) MoM is 100% reliable for added delay greater than 40ms whereas MoG achieves 80% reliability. When the added delay is reduced to 20ms both methods converge to 80% reliability and continue to experience the same reliability value as the delay is reduced further. For a shorter observation window (10s) a similar trend can be observed; at 50ms added delay MoM achieves 78% reliability with MoG achieving 68%. As added delay is reduced, both methods convergence in performance so that MoG achives 42% reliability at 20ms compared with 36% for MoM. As added delay is reduced to 10ms, both methods exhibit similar performance. With a short observation window (1s) we observe the same trend as with longer windows but with much reduced reliability. With 50ms added delay MoG has a reliability of 37% while MoM achieves 40%. With 30ms added delay MoG performs best with 31% reliability compared to 28% with MoM. This trend continues with MoG being 27% reliable at 10ms and MoM being 23%. In this case both estimators achieve very similar performance.

From the above we note that with a long observation window and with an added delay greater than 20ms the MoM provides the most reliable method of bottleneck-link detection. If the added delay is reduced to less than 20ms and the observation window shortened to 10s or less then the MoG achieves a similar performance. Practically, this implies that in a network where delays on a link are within 20ms of each other and only a short observation period is available then using a either method would be equally likely to provide correct detection of the bottleneck link.

3.3. Estimator Rates

One consideration for both methods is the number of probe packets required to perform reliable bottleneck-link detection. Here, we consider the effect of reducing the probe traffic rate on both methods studied above; reduction of the probe rate improves efficiency by congesting the links with fewer probe packets, however, this is at the cost of accuracy.

To evaluate this trade-off we use the topology and estimation methods shown previously but adjust the simulation parameters such that Links 1, 2 & 3 have 10ms and Link 4 has 50ms added delay. We use 100 realisations of a 10s observation window to remain consistent with previous results.

From Fig 4 we see that with an estimator rate of 1Kb/s (half that used in the previous scenario) the MoM achieves a reliability of 94% with MoG achieving 92%. As the rate is reduced to 0.7Kb/s the MoM reliability is reduced to 87% whilst the MoG achieves only 64%. As the rate is further reduce to 0.5Kb/s, a quarter of the original, the reliability of the MoG falls to 49% whilst the MoM has fallen to 80%. This suggests that a MoM approach is preferable when the number of probe packets is low. This would appear consistent with Fig 3, reducing the estimator packet rate and reducing the observation window have the effect of reducing the data

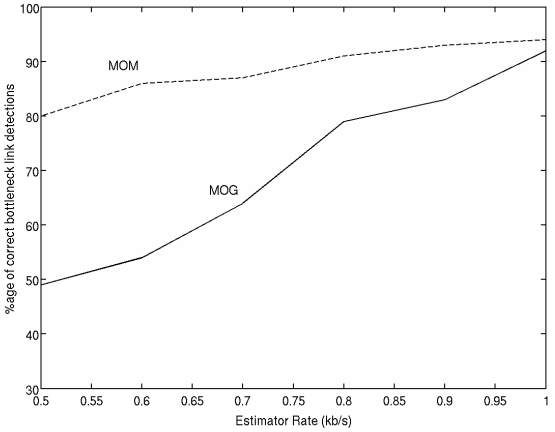

Fig. 4. Estimator Rate against Correct Detection Rate for both methods with a 10s observation window

available to the estimator which results in a lower probability of correct bottleneck link detection. The corollary also applies, the probability of correct bottleneck detection can be improved by either raising the estimator packet rate or increasing the observation window to increase the data available to the path-level estimator.

3.4. Computational Complexity

We compare the computational intensity of each method by considering the number of Multiply (MULT) and Add (ADD) operations required:

	MoM	MoG
MULT	$2NPt + Pt + PL - L$	$PN + 2P + 3PL$
ADD	$NPt - Pt + PL - L$	$2PN + 2PL - 2L$

Table 2. Formulae for number of operations required

In Table 2 we present equations for the complexity of the data processing part both methods, (ie excluding the sampling process) in terms of the simulation parameters. We see the complexity of both methods scales by the number of samples, N, but that MoM also scales by CGF parameter, t. Here, and in the scenarios described in Section 3.1, P, L & t are 5, 4 & 20 respectively. In Table 3 and Fig 5 we see the number of operations required in the scenarios previously mentioned where we observe that MoG requires an order of magnitude fewer operations than MoM. However, this comes at the expense of the reliability of bottleneck-link detection.

Fig 5 shows graphically how the number of operations quickly scales as the size of the network increases. Here we have assumed the ratio L/P has remained constant at 0.8 as in the previously defined scenarios.

	MoM	MoG
MULT	600116	15070
ADD	299916	30032

Table 3. Numerical results for number of operations required

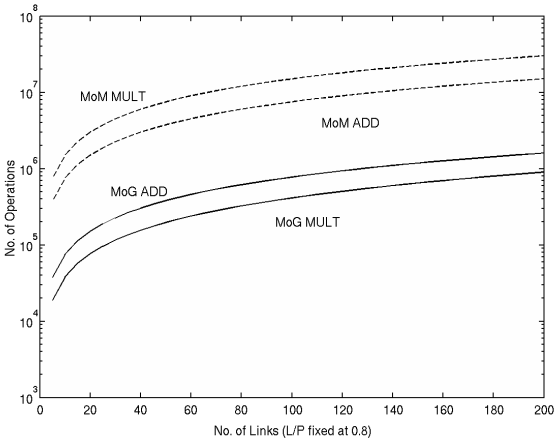

Fig. 5. Number of Operations against Number of Links in network for a fixed value of L/P

Finally, we consider the complexity of inverting H as mentioned in Section 2.1. In [10] it is estimated that inverting a matrix of size $N * N$ takes a number of operations of order N^3. In our scenario, P and L are similar in size so we estimate the operation will be of similar complexity, around $O(L^3)$. As this is a wired scenario, we assume the inversion takes place once as the topology remains fixed throughout; however, we note that were it a wireless scenario with mobile nodes then this would be a more significant contribution to overall complexity.

4. CONCLUSION & FUTURE WORK

In this paper we have presented a comparison of two methods of delay estimation and bottleneck-link discovery for use in wired network tomography. We have shown that the parametric estimation (MoG) method provides performance comparable with the CDF estimation (MoM) method for a short observation window with a reasonable probe-packet rate. We have seen that MoG is less reliable than MoM for low probe-packet rates. In both cases, MoG has the advantage of a reduced computational complexity. We have also shown that performance in both methods can be improved by increasing the length of the observation window.

In future we will consider more methods, of both parameter estimation and CDF estimation types. We imagine that with an accurate model, a parametric method would be the most reliable even with a low probe-packet rate. For a higher probe-packet rate, greater accuracy can be obtained with a CDF estimation method, however, this is at the cost of increased computational complexity.

5. REFERENCES

[1] Y. Vardi, "Network tomography: Estimating source-destination traffic intensities from link data," *J. Amer. Stat. Assoc*, vol. 91, no. 433, pp. 365–377, 1996.

[2] A. Coates, A. O. H. III, R. Nowak, and B. Yu, "Internet tomography," *IEEE Signal Processing Mag.*, vol. 19, no. 3, pp. 47–65, May 2002.

[3] M.-F. Shih and A. Hero, "Unicast inference of network link delay distributions from edge measurements," in *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*, vol. 6, Salt Lake City, UT, May 2001, pp. 3421–3424.

[4] Y. Xia and D. Tse, "Inference of Link Delay in Communication Networks," *IEEE J. Select. Areas Commun.*, vol. 24, no. 12, pp. 2235–2248, Dec. 2006.

[5] R. Caceres, N. G. Duffield, J. Horowitz, and D. F. Towsley, "Multicast-based inference of network-internal loss characteristics," *IEEE Trans. Inform. Theory*, vol. 45, no. 7, pp. 2462–2480, Nov. 1999.

[6] M. J. Coates and R. D. Nowak, "Network tomography for internal delay estimation," in *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*, vol. 6, Salt Lake City, UT, May 2001, pp. 3409–3412.

[7] N. G. Duffield, F. L. Presti, V. Paxson, and D. Towsley, "Inferring link loss using striped unicast probes," in *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2, Anchorage, AK, Apr. 2001, pp. 915–923.

[8] Multicast-based inference of network-internal characteristics (MINC). [Online]. Available: http://gaia.cs.umass.edu/minc/

[9] UCL/VINT/LBNL. network simulator ns (version 2). [Online]. Available: http://www.isi.edu/nsnam/ns/

[10] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*, 2nd ed. Cambridge, England: Cambridge University Press, 1992, ch. 2.