A COMPARISON OF SOME BOTTLENECK-LINK DETECTION METHODS FOR NETWORK TOMOGRAPHY

Nick Johnson †, John Thompson †, Steve McLaughlin †and Francisco J. Garcia ‡

†
Institute for Digital Communications
Joint Research Institute for Signal & Image Processing
School of Engineering
The University of Edinburgh
Edinburgh, EH9 3JL, UK
{Firstname . Surname}@ed.ac.uk

‡
Agilent Laboratories, (Scotland)
South Queensferry
Edinburgh, UK

Frankie_ garcia@agilent.com

ABSTRACT

Network tomography offers a useful method to identify internal problems in a network using data which can be obtained at the network's edge. Provided the topology of the network is known then it is possible to recover from the edge measurements some properties of internal links of the network which may not be accessible for any number of reasons - cost, ownership, physical location etc. In this paper we introduce two new estimation algorithms based on the Pearson type-1 distribution and compare these with existing estimator and detector algorithms used to find the bottleneck link in a wired network, that is, the link experiencing the highest delay.

1. INTRODUCTION

Computer networks are growing in both size and complexity and with the evolution of the internet can be connected in a near infinite number of ways. Typically, the user is situated at the edge of such a network with the resources they wish to access at another edge; thus, the route taken by data travelling between user and resource can be long and complex. Data packets will interact with other traffic on some or all of the route which will have an affect on their statistical properties such as latency (delay), loss rate and interarrival time. It is desireable to measure the network to detect anomalous values of these statistics; network tomography, introduced in [1] is one method of doing this.

In this work, we concentrate on latency-based tomography, in particular finding the network link with highest latency. We define a network link as a connection between two nodes or routers in a network and a route or path as a connected set of links, this is illustrated in Figure 1. We estimate the route-level delay distribution from measurements available at the route or path level, convert this to an estimate of the link-level distribution and then detect which link is most probably the one with highest delay (which we refer to as the bottleneck-link). There are many approaches to this problem, some of which use parametric distributions for link and path estimates such as exponential and exponential mixture distributions [2] or Gaussian mixture distributions [3]. Alternatives use EM based approaches [4] whilst some are based upon particle filters [5]. Some of the most recent and potentially most efficient, in terms of volume of data required to detect any bottleneck, are based upon compressed sensing [6]. Our contribution is in extending two existing estimation algorithms using the Pearson type-1 distribution to form two new estimation algorithms which may offer a computational saving over the originals.

The remainder of this paper is organised as follows: in Section 2 we introduce our system model, in Section 3 we introduce the estimation algorithms we are going to compare and in Section 4 we introduce detection algorithms to accompany them. In Section 5 we introduce the parameters used for the test and show some selected results before finally, in Section 6 presenting some conclusions.

2. SYSTEM MODEL

It is possible to describe tomography using some basic algebra; here, we attempt to remain consistent with other works [1], [7] in our nomenclature. We refer to the path-level delay measurements as Y, the link-level delay data we cannot measure but wish to estimate as X and the routing matrix (which describes the relationship between X and Y) as H. The paths in any network are numbered from 1 to P while the links are numbered from 1 to L. Using this notation, we can think of a network as:

$$Y = H \times X \tag{1}$$

This implies that the P routes (or paths, the terms are often used interchangeably) in a network are simply an interconnected set of L links: H, the routing matrix, describes how the links are connected to form the paths using a 1 to indicate the the link is part of the route and a 0 to indicate otherwise. Since our desire is not to gather information on Y (which we can measure directly) but on X, we invert our equation:

$$X = H^{-1} \times Y \tag{2}$$

We can see that this is the well known least squares (LS) method applied to finding X where H^{-1} is the L2 pesudo-inverse of H. We seek to find an estimate of X as a weighted sum of the contributions from each Y with the weights coming from H^{-1}. We define h_{ij} as the weight assigned to the contribution to link j from path i.

It is perhaps easiest to illustrate this using a small example so consider a networl which has a topology such as that shown in Figure 1 with $P = 5$ and $L = 4$. Link 1 and link2 form path 1 so the routing matrix has a 1 in the first two columns of

the first row: the columns of the routing matrix correspond to the links while the rows correspond to the paths. Equation 3 shows the routing matrix (H) while equation 4 shows it's pseudo-inverse (H^{-1}).

$$H = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (3)$$

$$H^{-1} = \begin{bmatrix} 0.50 & 0.50 & 0 & -0.50 & -0.50 \\ 0.25 & -0.25 & 0 & 0.25 & 0.75 \\ -0.50 & -0.50 & 1 & 0.50 & 0.50 \\ 0.25 & 0.75 & 1 & 0.25 & -0.25 \end{bmatrix} \quad (4)$$

The tomography problem presented above can be reduced in it's most basic format to this: trying to estimate the link delay CDF (X) using measurements taken at the route level (Y) and knowledge of how the network is constructed (H). This can be used to determine the link with the highest delay which can be useful for monitoring and control applications.

3. ESTIMATION ALGORITHMS

In this section we introduce four algorithms to estimating the statistical properties of the links based on the data acquired at the path level; the results are used with a complementary detection algorithm described in Section 4 and the performance shown in Section 5.

3.1 Gaussian Approximation

The use of a single Gaussian distribution to model link delay was suggested in [2] although the authors believed it to have problems with identifiably. We also treat the path delay as having a Gaussian distribution so that the mean and variance are estimated from N path data and subsequently transformed to provide an estimate of the mean and variance of the link using LS with weights coming from $|h_{ij}|^2$. Equations (5) & (6) show the estimation of the mean and variance for path i respectively while equation (7) shows the estimated distribution for link j.

$$\widehat{\mu_{Y_i}} = \frac{1}{N}\sum_{k=1}^{N} Y_{ik} \quad (5)$$

$$\widehat{\sigma^2_{Y_i}} = \frac{1}{N}\sum_{k=1}^{N} (Y_{ik} - \widehat{\mu_{Y_i}})^2 \quad (6)$$

We treat the variance as a noise process and so square the weights ($|h_{ij}|^2$) to preserve the positivity of the link estimate.

$$X_j = \mathcal{N}(\sum_{i=1}^{P} \widehat{\mu_{Y_i}} \times h_{ij}, \sum_{i=1}^{P} \widehat{\sigma^2_{Y_i}} \times |h_{ij}|^2) = \mathcal{N}(\widehat{\mu_{X_j}}, \widehat{\sigma^2_{X_j}}) \quad (7)$$

Algorithm 1 GA estimation algorithm

1: **for** $i = 1$ to P **do**
2: fit single Gaussian to path, i, using equations (5) & (6)
3: **end for**
4: **for** $j = 1$ to L **do**
5: estimate PDF of link j using path estimates 1 to P using LS as per equation (7)
6: **end for**

3.2 Method Of Moments (MOM)

In [1] and [7] the authors estimate the Cumulant Generating Function (CGF) of the distribution of delay on each path from individual measurements with a method-of-moments (MOM) estimator. 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 $Y_{ik}, k = 1...N$,

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

Then we use LS to obtain a link-level estimate of the CGF.

$$\widehat{K_{X_j}} = \sum_{i=1}^{P} h_{ij} \times \log(\widehat{M_{Y_i}}) \quad (9)$$

Algorithm 2 MOM estimation algorithm

1: **for** $i = 1$ to L **do**
2: estimate CGF of path, i, using equation (8)
3: **end for**
4: **for** $j = 1$ to L **do**
5: estimate CGF of link j using path estimates 1 to P using LS as per equation (9)
6: **end for**

3.3 Pearson type-1 Distribution (PRS)

With GA we use the first two moments to model the data but we suspect that this might not allow enough flexibility to give an accurate model so we use a Pearson type-1 distribution to make use of the first four moments. The algorithm is similar to GA but with the addition of skewness and kurtosis which we recall in equations (10) and (11) respectively.

$$\widehat{\mu_3} = \frac{1}{N}\sum_{k=1}^{N} E((Y_{ik} - \mu_{i,1})^3)/\sigma^3 \quad (10)$$

$$\widehat{\mu_4} = \frac{1}{N}\sum_{k=1}^{N} E((Y_{ik} - \mu_{i,1})^4)/\sigma^4 - 3 \quad (11)$$

LS is applied to the four estimates for each path as in GA and similarly results in a set of estimates for a Pearson distribution for each link. We recall that the Pearson has a condition in that $\mu_4 \geq \mu_3^2 + 1$ which we find may not always occur due to the transformation in LS; in this situation we modify the values of μ_3 so that the condition is satisfied.

3.4 Pearson - Method Of Moments (P-MOM)

The problem with MOM is that it does not produce an estimate of the CDF (or PDF) of the delay for each link which

Algorithm 3 PRS estimation algorithm

1: **for** $i = 1$ to P **do**
2: fit a Pearson type-1 distribution to path i, using equations (5), (6), 10 & 11.
3: **end for**
4: **for** $j = 1$ to L **do**
5: estimate CDF of link j using path estimates 1 to P using LS in a similar manner to equation (7)
6: **end for**

is preferable to the CGF as it may give more insight into the operation of the network. To overcome this, we fit a Pearson type-1 distribution to the first four parameters estimated by MOM.

Algorithm 4 P-MOM estimation algorithm

1: **for** $i = 1$ to L **do**
2: estimate CGF of path, i, using equation 8
3: **end for**
4: **for** $j = 1$ to L **do**
5: estimate CGF of link j using path estimates 1 to P using LS as per equation 9
6: **end for**
7: **for** $j = 1$ to L **do**
8: generate Pearson type-1 distribution using cumulants from CGF for each link j
9: **end for**

4. DETECTION ALGORITHMS

To detect the bottleneck-link we employ a detection algorithm compatible with the estimator output. Both detection algorithms used here require the a-priori selection of a variable, δ, which is an educated guess at the value of delay. The choice of δ is empirical and detection accuracy is often dependant on it; both major limitations.

4.1 CDFmax

We evaluate the link CDFs at a fixed value of δ and call this P_j and pick as bottleneck the link with lowest P_j. This relies on a good CDF estimate to achieve reliable detection but is suitable for any of the parametric algorithms - ie GA, PRS and P-MOM.

$$P_j = \arg\min_j \left(cdf_j(\delta) \right); \quad j\varepsilon\{1,2,\ldots,L\} \qquad (12)$$

4.2 Chernoff Bound

We impose a Chernoff upper-bound on the link CGFs and select as bottleneck the link with the highest probability (P_j) of exceeding the delay threshold (δ).
In [1] and [7] this is expressed as:

$$P_j = P(X_j \geq \delta) \leq e^{-t\delta} E[e^{tX_j}] \qquad (13)$$

5. SELECTED SIMULATION RESULTS

5.1 Simulation Setup

To test all the methods (where a method is a combination of estimation and detection algorithms, named after the estimator) we use an ns2 [8] simulation to model a wired network with unicast probe-path traffic. The topology of the 5-node network is shown in Figure 1 while the 10 node network has a larger but similar structure: the key parameters are shown in Table 1. Background traffic on each link is formed by combining exponentially-distributed constant-bit-rate UDP and TCP traffic sources similar to those in [1]. The rates of the UDP sources are chosen to ensure each link is between 70% and 80% utilized; the TCP sources adjust their rate to achieve maximum throughput ensuring links operate at peak capacity. This is important because we want to ensure that packets on the network experience some delay due to congestion. The delay is manifested as queueing and processing time at each node. In addition, on each link we add a delay to each packet so that one link has a delay higher than the others for our methods to detect. This ensures we have a scenario where we know the bottleneck and can control the degree of detection difficulty.

Parameter	Value
Bottleneck delay (5 node)	120ms, 150ms
Bottleneck delay (10node)	150ms, 200ms, 250ms
Normal link delay (5 node)	10ms, 80ms, 100ms
Normal link delay (10 node)	100ms
Link bandwidth	1 Mb
Simulation time	5000s
Number of paths, P (5 node)	5
Number of paths, P (10 node)	12
Number of links, L (5 node)	4
Number of links, L (10 node)	9
CGF parameter, t	20
Number of probe packets, N	25000
Probe packet rate	2 Kb/s
Probe packet size	40 Bytes

Table 1: Key Simulation Parameters

5.2 Discussion of Results

We compare the performance of the estimator and detector combinations in terms of detection accuracy and computational complexity in six different scenarios:

- A 5 node topology with 20ms separation (Figure 2a)
- A 5 node topology with 50ms separation (Figure 2b)
- A 10 node topology with 50ms separation (Figure 3a)
- A 10 node topology with 150ms separation (Figure 3b)
- A 10 node topology with 250ms separation (Figure 3c)

In Figure 2a PRS outperforms GA in terms of peak accuracy by about 20%, however both methods show sensitivity to choice of δ; this may indicate information loss when using two moments in GA. MOM is less sensitive to δ than P-MOM although both offer a similar level of accuracy of between 50 and 60% which we assign to the MOM-based parameter estimation.

In Figure 2b the separation has increased: GA now outperforms PRS, the peak accuracy is greater by 6% while the range of δ over which it has high accuracy has increasedr. MOM and P-MOM show similar performance with the difference being slightly less than in Figure 2a.

If the separation increases to a larger value, ie 150ms, then all methods converge to 100% accuracy. This is not unexpected as the separation is high (greater than twice the delay of the 2nd worst link) and it should be easy to detect a bottleneck even if the estimation is poor.

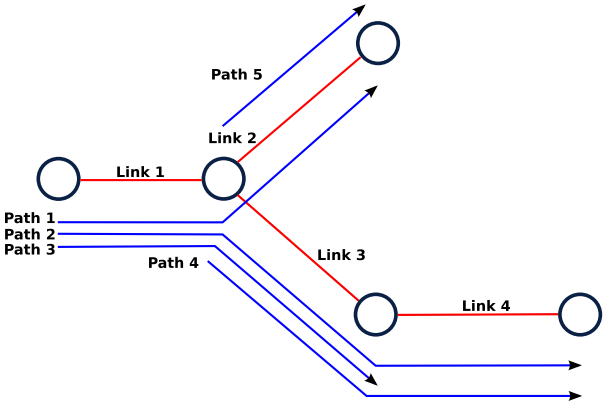
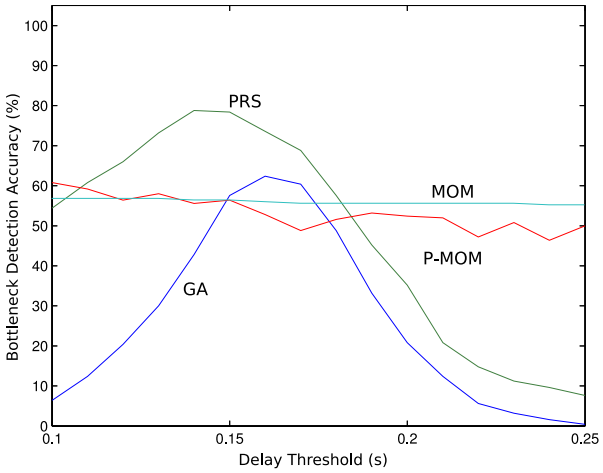
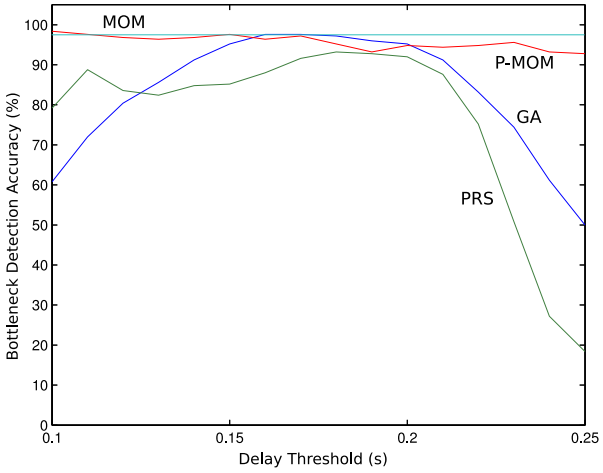


Figure 1: Network topology showing paths and links, originally from [1]

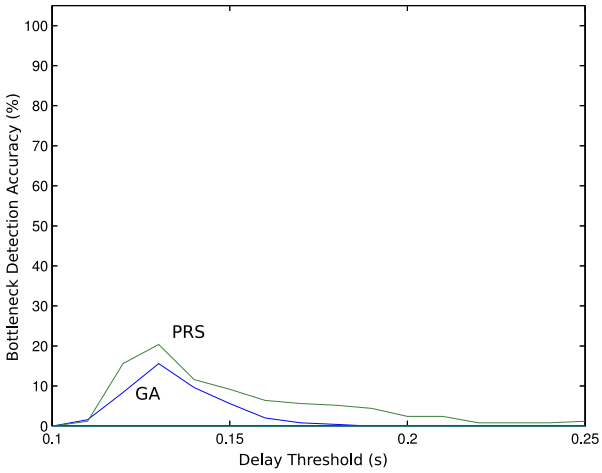


(a) 20ms separation

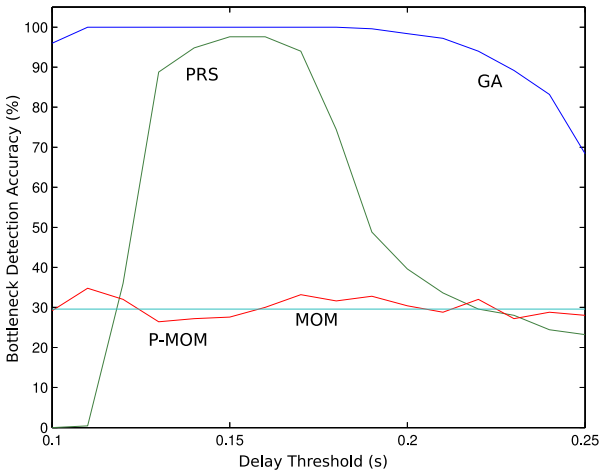


(b) 50ms separation

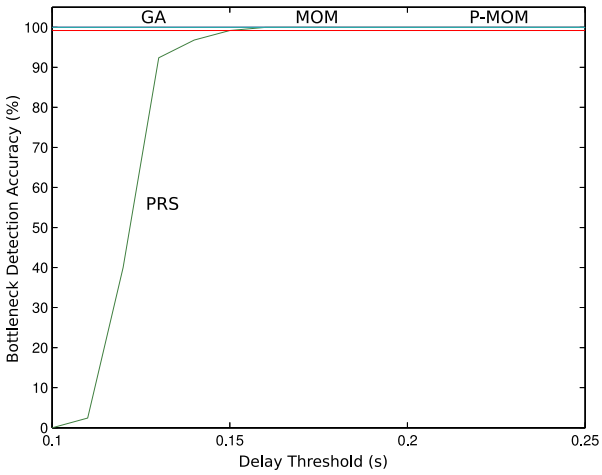
Figure 2: Detection Accuracy against choice of delay threshold (δ) for two 5 node scenarios



(a) 50ms separation



(b) 100ms separation



(c) 250ms separation

Figure 3: Detection Accuracy against choice of delay threshold (δ) for three 10 node scenarios

Algorithm	MULT
GA	$L(87+3P)+P(N+2)$
PRS	$P(11N+5L+2)+165L$
MOM	$Pt(2N+1)+L(5t+2)$
P-MOM	$167L+4P(N-1)$

Algorithm	ADD
GA	$30L+L(P-1)+3PN$
PRS	$8PN+4L(P-1)+52L+(L-1)!$
MOM	$PtN+LA+(L-1)!$
P-MOM	$L(P+52)+4PN+(L-1)!$

Table 2: Formulae for number of MULT and ADD operations required for estimation/detection of 20s of data in the 5 node network.

Algorithm	MULT	ADD
GA	5418	15136
PRS	55770	40278
MOM	200508	116006
P-MOM	20648	20234

Table 3: Numerical results for the formulae in Table 2 using values from Table 1 for the scenario in Figure 2a with $N = 1000$.

Figure 3a shows the performance of the methods in the 10 node scenario. Both MOM and P-MOM are unable to correctly detect the bottlneck while GA and PRS exhibit poor performance although PRS is the more accurate. As the topology scales, the separation required for a fixed level of accuracy increases.

In Figure 3b the accuracy of GA is comparable to Figure 2b: the range of PRS is narrower and peak accuracy is slightly less than GA. MOM and P-MOM exhibit similar accuracy of around 30% with P-MOM having some variance as in the 5 node scenarios.

In Figure 3c the accuracy of GA, MOM and P-MOM has increased to the maximum which is expected given the large separation. Interestingly, the PRS response has a lower tail similar to that in Figure 3b showing sensitivity to δ.

Table 2 shows the number of multiplication (MULT) and add (ADD) operations required to perform one estimation and detection on a block of data (around 20s of data) using the scenario in Figure 2a. GA is the most computationally efficient as it has a low number of parameters and scales with the number of samples, N. PRS scales with N but has a more complex CDF and a greater number of parameters than GA. MOM scales with N and t which increases the complexity for even a small value of t. P-MOM requires t equal to 4 so offers a complexity saving compared to MOM. Table 3 expresses this numerically with the parameters as in Table 1 but with $N = 1000$. This illustrates that MOM is the least computationally efficient followed by PRS. Crucially we see that P-MOM combines the efficient estimation of MOM with the efficient output evaluation of PRS requiring one fifth the number of ADDs and one tenth the number of MULTs of MOM.

With large separation, all methods are able to correctly identify the bottleneck in a network. As separation decreases, the parametric methods (GA and PRS) appear more robust, however they are sensitive to choice of δ making them unsuitable in an environment where little is known about the normal operating conditions of the network. A non-parametric method such as MOM has reduced sensitivity to δ but its output may not be compatible with all types of detection algorithm. A hybrid approach, P-MOM, which uses the Pearson distribution as output sacrifices some of the accuracy of MOM but gains reduced sensitivity and decreased computational cost.

6. CONCLUSION

In this paper we compared four network tomography methods, two of which make use of the Pearson distribution and are introduced here, to perform bottleneck-link detection. We saw that a non-parametric algorithm (MOM) provides consistent, reliable performance which is not constrained by an *a-priori* choice of delay threshold. However it was not as robust as a parametric algorithm (GA and PRS) in low separation scenarios and was computationally expensive. Robustness has to be carefully traded with computational cost when considering overall suitability, especially in a real-time environment and we conclude that it may be preferential to sacrifice robustness for a reduction in compute time (GA). Where sensitivity to δ is a concern and a CDF output is desireable then our hybrid method, P-MOM, offers a solution which combines the flexibility of PRS with the consistent performance of MOM.

REFERENCES

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

[2] Y. Xia and D. Tse, "Inference of link delay in communication networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 12, pp. 2235–2248, 2006.

[3] N. Johnson, J. Thompson, S. McLaughlin, and F. Garcia, "A comparison of delay estimation & bottleneck-link detection methods for network tomography," in *Proceedings of the 8th IMA conference on Mathematics in Signal Processing*, Cirencester, UK, Dec 2008.

[4] Y. Sun, D. Li, and H. Sun, "Network Tomography and Improved Methods for Delay Distribution Inference," in *The 9th International Conference on Advanced Communication Technology*, vol. 2, Gangwon-Do, Feb. 2007, pp. 1433–1437.

[5] M. J. Coates and R. D. Nowak, "Sequential monte carlo inference of internal delays in nonstationary data networks," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 366–376, 2002.

[6] M. Coates, Y. Pointurier, and M. Rabbat, "Compressed Network Monitoring," in *SSP '07. IEEE/SP 14th Workshop on Statistical Signal Processing*, Madison, WI, USA, Aug. 2007, pp. 418–422.

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

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