

# A Flexible Tomography Approach for Queueing Delay Distribution Inference in Communication Networks

Gábor Simon, Péter Hága, Gábor Vattay and István Csabai

Collegium Budapest Institute for Advanced Study, H-1014, Szentháromság u. 2, Budapest, Hungary

E-mail: {gsimon, haga, vattay, csabai}@colbud.hu

## Abstract

*We present in detail a flexible method for inferring internal queueing-delay distributions in a network, by performing correlated unicast packet-pair measurements. The method is based on the quantization of the measured end-to-end delays, maximum-likelihood estimation via the expectation maximization (EM) algorithm, and making numerical deconvolutions via the non-negative least squares (NNLS) algorithm. We have tested the inference method in realistic network simulations as well as in real local area network (LAN) measurements using different scenarios and found impressive agreement between the estimated and real queueing-delay distributions.*

## 1 Introduction

Recent advances in the field of network tomography attracted considerable attention of the networking community. The tomography approach is a special class of active-probing measuring techniques, that aims at determining relevant characteristics, such as delay statistics [5, 6] and packet loss rates [4, 9] in the interior of a network, solely from end-to-end measurements performed between the edges. Using such a strategy improves significantly on the utilization of Internet measuring infrastructures, that otherwise could reveal only end-to-end network conditions between the participating hosts.

Initially network tomography techniques were developed for the use with multicast probes [8, 13], but later these approaches were extended also to the unicast case. The idea is to use back-to-back packet pairs or multiples, where each packet contained in the pair or multiple is destined to different receivers. Under certain conditions these unicast streams also show the nice correlation properties present in the multicast probing technique that is the key to resolve the internal characteristics from end-to-end measurements. With this extension network tomography became a generally applicable tool that has the advantage that it does not rely on the cooperation of the internal network

elements more than just to store and forward the incoming probe packets.

In general a tomography measurement made from a single source admits the determination of the delay statistics and loss-rates on each segment of the underlying measurement tree spanned by the paths of the different probes. Depending on the number of receivers and the topology the measurable internal portion may cover a considerable part of the whole network, which can be further extended by performing the measurements from multiple sources [2].

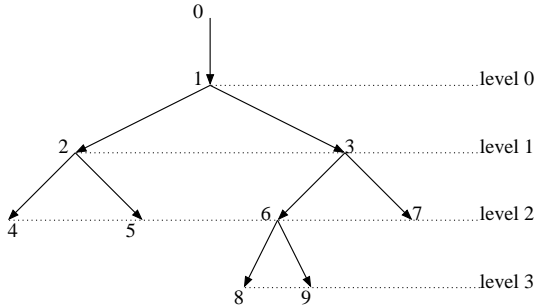
In this paper we are concerned with the determination of internal queueing delay distributions from end-to-end delay measurements, given prior knowledge of the topology between the source and the receivers. Our motivation to study queueing delay distributions stems from the fact that it carries information about traffic properties and the state of congestion on the measured path. By resolving the queueing delay distributions from end-to-end measurements one can draw a map of congestion of the network segments included in the measurement tree, and thus identify highly congested segments.

To perform queueing delay tomography we need time-series of end-to-end queueing delays that can be obtained by subtracting the minimum from the measured delay time-series. Here we make the implicit assumption that in any measurement of the end-to-end delays some probes will experience no queueing and thus provide the constant part of the delay. Measurement studies indicate that in the current Internet link utilizations are low enough so that the above assumption is justified [11, 12].

The rest of the paper is organized as follows. In Section 2 we provide the detailed description of the queueing delay inference method. Section 3 evaluates its performance in realistic ns-2 simulations of different network scenarios. In the subsequent Section 4 we describe our high-precision measuring infrastructure and evaluate the performance of the inferring method in a real measurement. Finally the conclusion is given in Section 5.

## 2 Queueing delay inference

### 2.1 Labeling and notations



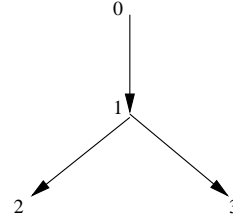
**Figure 1. The labeling of the different nodes and network segments of the measurement tree.**

To achieve a unique labeling of the different network segments of the measurement tree we use the arrangement illustrated in the example of Fig. 1. The source node is situated on the top of the hierarchy, labeled with “0”, while branching nodes are put on a given level of the hierarchy according to their “distance” from the source. This distance is measured by the number of branching nodes a probe packet must pass to reach the target. Different nodes on the same hierarchical level are enumerated in the incremental order from the left to right. Using this scheme one can identify each node by a unique index  $i \in (0, 1, \dots, N_{br} + N_R)$ , where  $N_{br}$  and  $N_R$  are respectively the number of branching and receiver nodes in the measurement tree. Consequently the different segments of the measurement paths, indicated by arrows in Fig. 1, can also be identified uniquely by the indices of the target nodes. We denote the one-way delay experienced by the  $n$ -th probe on the  $i$ -th segment by  $\hat{X}_i(n)$ , and the corresponding end-to-end delay from the source to the  $i$ -th receiver node by  $\hat{Y}_i(n)$ . The queueing delays associated to these quantities are denoted by  $X_i(n)$  and  $Y_i(n)$ . By dropping the  $n$  index from  $\hat{X}_i$ ,  $\hat{Y}_i$ ,  $X_i(n)$  and  $Y_i(n)$  we indicate the time-series of the respective quantities.

### 2.2 Inference in the two-leaf tree

Here we describe the method of queueing-delay distribution inference in the two-leaf tree illustrated in Fig. 2. The known quantities are the time series of end-to-end queueing delays  $Y_2$  and  $Y_3$ , obtained by minimum filtering the  $\hat{Y}_2$  and  $\hat{Y}_3$  time-series

$$Y_2(n) = \hat{Y}_2(n) - \min(\hat{Y}_2), \quad Y_3(n) = \hat{Y}_3(n) - \min(\hat{Y}_3). \quad (1)$$



**Figure 2. The two-leaf tree, an example of the smallest network topology where the queueing delay distributions on the internal segments can be resolved from end-to-end measurements.**

These are in turn measured by a big number of packet pairs, where the packets in a pair are sent back-to-back from the source and are destined to the two distinct receivers. We count only those pairs where each of the packets has reached its destination, and assign the number of successful pairs by  $N$ . Using the labeling introduced earlier we have the trivial relations

$$Y_2(n) = X_1(n) + X_2(n), \quad Y_3(n) = \hat{X}_1(n) + X_3(n), \quad (2)$$

where  $\hat{X}_1(n)$  stands for the queueing delay on segment 1, experienced by that particular probe in the  $n$ -th pair which was destined toward the receiver node (3). If the time separation of the packets in a pair is smaller than the time-scale at which queue sizes change considerably, then one may expect that these probes will experience essentially the same delays on the common segment ( $X_1 = \hat{X}_1$ ), thus from this point we shall not distinguish these quantities and use only  $X_1$ . The goal is to estimate the distribution of the unknown  $X_1$ ,  $X_2$ , and  $X_3$  time-series, based on the knowledge of  $Y_1$  and  $Y_2$ .

This can be achieved by introducing the quantized versions of the queueing delays. By using the quantization rule  $Y_i^d(n) = jq$ , if  $(j - 1/2)q < Y_i(n) \leq (j + 1/2)q$ , one can map the end-to-end time-series  $Y_i$  to their quantized versions  $Y_i^d$ , that take values from the set  $(0, q, 2q, \dots, Bq)$ . With the similar quantization we can also introduce the quantized versions of the unknown time-series of  $X_i^d$ . The quantization parameters  $q$  and  $B$  have to be chosen so that  $\max(Y_2, Y_3) < (B + 1)q$ , which ensures that all the possible values of  $Y_i$  and  $X_i$  will be included in one of the bins defined by the quantization rule, and thus probabilities can be assigned to the bins. We denote the probability of a queueing delay falling in the  $j$ -th bin on the  $i$ -th segment by  $P_{i,j}$ , and for each  $i \in (1, 2, 3)$  we have  $\sum_{j=0}^B P_{i,j} = 1$ . These probabilities are estimated by

$$P_{i,j} = \frac{m_{i,j}}{\sum_{j=0}^B m_{i,j}} = \frac{m_{i,j}}{N}. \quad (3)$$

where  $m_{i,j}$  stands for the number of times a probe experienced the quantized queueing delay  $X_i^d = jq$  out of all the  $N$  successful pairs. The reason behind the introduction of the discrete quantities above was that using the  $P_{i,j}$  values one can express the probabilities of finding the quantized end-to-end queueing delays in a given bin, which can be estimated from the measured data. The probability of observing a quantized packet-pair delay of  $(Y_1^d = lq, Y_2^d = mq)$  is given by the convolution

$$P(l, m) = \sum_{k \in H} P_{1,k} P_{2,(l-k)} P_{3,(m-k)}, \quad (4)$$

where the set  $H$  is given by  $\{B \geq k \geq 0\} \cap \{B \geq (l-k) \geq 0\} \cap \{B \geq (m-k) \geq 0\}$ . Finally the probabilities  $P_{i,j}$  can be determined from the property that their right value maximizes the log likelihood-function

$$\log \mathcal{L} = \sum_{n=1}^N \log(P(Y_1^d(n), Y_2^d(n))), \quad (5)$$

where the probabilities  $P(Y_1^d(n), Y_2^d(n))$  are given by equation (4) evaluated at the known quantized end-to-end queueing delay pairs. Although the log-likelihood-function (5) can not be maximized analytically, there are several numerical procedures to accomplish this goal. Reference [5] suggests to apply the expectation-maximization (EM) algorithm [10], which we briefly review here without derivation, that can be found [5].

One starts by assigning initial values for all the  $P_{i,j}$  probabilities. These are then used to calculate estimates of the  $m_{i,j}$  quantities by

$$m_{i,j} = \sum_{n=1}^N P(X_i^d(n) = jq | Y_1^d(n), Y_2^d(n)), \quad (6)$$

where  $P(X_i^d(n) = jq | Y_1^d(n), Y_2^d(n))$  stands for the conditional probability of  $X_i^d(n) = jq$ , given the values  $Y_1^d(n), Y_2^d(n)$  of the  $n$ -th packet pair delay. According to Bayes-law these conditional probabilities are

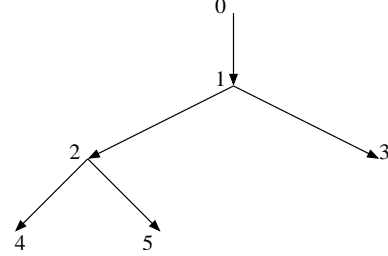
$$\begin{aligned} P(X_1^d = jq | kq, lq) &= \frac{P_{1,j} P_{2,(k-j)} P_{3,(l-j)}}{P(k, l)}, \\ P(X_2^d = jq | kq, lq) &= \frac{P_{1,(k-j)} P_{2,j} P_{3,(l-k+j)}}{P(k, l)}, \\ P(X_3^d = jq | kq, lq) &= \frac{P_{1,(l-j)} P_{2,(k-l+j)} P_{3,j}}{P(k, l)}. \end{aligned} \quad (7)$$

The values of  $m_{i,j}$ , calculated from equations (6,7), provide the improved estimates  $\hat{P}_{i,j}$  through equation (3). This process is iterated until a suitable convergence criterion is met. We terminate the iterations if for all  $i, j$

$$\Delta_{i,j} \leq \epsilon/B, \quad (8)$$

where  $\Delta_{i,j} = (\hat{P}_{i,j} - P_{i,j})/P_{i,j}$ , and  $\epsilon$  is a chosen small number, e.g. 0.0001.

## 2.3 Inference in larger networks



**Figure 3. An example of a measurement tree with 3 receiver nodes.**

The inference of queueing delay distributions in larger topologies can be accomplished using the algorithm developed for the two-leaf tree and making additional numerical deconvolutions. To resolve all the network segments with a minimal approach one must perform packet-pair measurements in a way that all the branching points and all the receivers are included at least in one of the measurements.

Consider for instance the topology shown in Fig. 3. Performing the measurements to the receiver pairs of (4,5) and (5,3) produces the distributions of  $X_4, X_5, conv(X_1, X_2)$  and of  $conv(X_2, X_5), X_3, X_1$ . Here  $conv()$  stands for the convolution of the distributions indicated in its arguments. Thus by performing numerical deconvolution e.g. via the NNLS algorithm [7], the remaining unknown distribution  $X_2$  can be recovered.

## 3 Simulation study

The inference method described in section 2 is based on some strict assumptions regarding the delay experienced by the probes. These include the spatial and temporal independence of the probe delays, and in particular it is assumed that the two packets in the same pair should experience *exactly* the same delays on the common segment of the measurement tree.

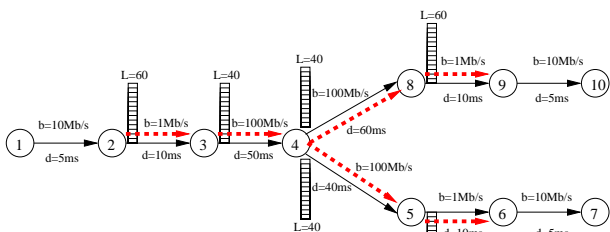
In reality all of these assumptions may be violated to some extent. For instance the presence of long-range dependent background traffic may cause slowly decaying temporal correlations of the probes on a given segment. Also in reality the two packets sent back-to-back may queue behind each other and develop a minimal inter-packet time  $\Delta t \approx P/b$  [3], where  $P$  is the size of the probe and  $b$  is the bottleneck bandwidth on the common portion of the measurement path. Additional increase in  $\Delta t$  may be

**Table 1. Background TCP traffic rates in the low and high utilization cases.**

low	2 → 3	3 → 4	4 → 5	5 → 6	4 → 8	8 → 9
$N_{TCP}$	40	0	0	40	0	40
$\langle ON \rangle$	20	0	0	10	0	15
high						
$N_{TCP}$	150	150	150	150	150	150
$\langle ON \rangle$	20	20	10	10	15	15

caused by background traffic that penetrates into the time gap between the probes.

To test the inference method against these weaknesses we performed packet level simulations with the ns-2 network simulator [1], that accurately implements the details of various traffic protocols and the properties of the most common network elements. Figure 4 depicts the studied

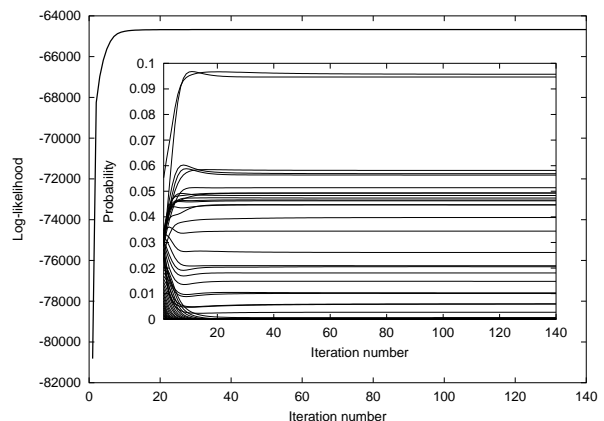


**Figure 4. Schematics of the network scenario built in ns-2.**

network scenario that is arranged in the topology of the two-leaf tree. The parameters of the network hops (link bandwidth  $b$ , propagation delay  $d$ , buffer limits of the drop-tail FIFO routers  $L$ ) are indicated in the figure, whereas all the non specified parameters are set to the default values.

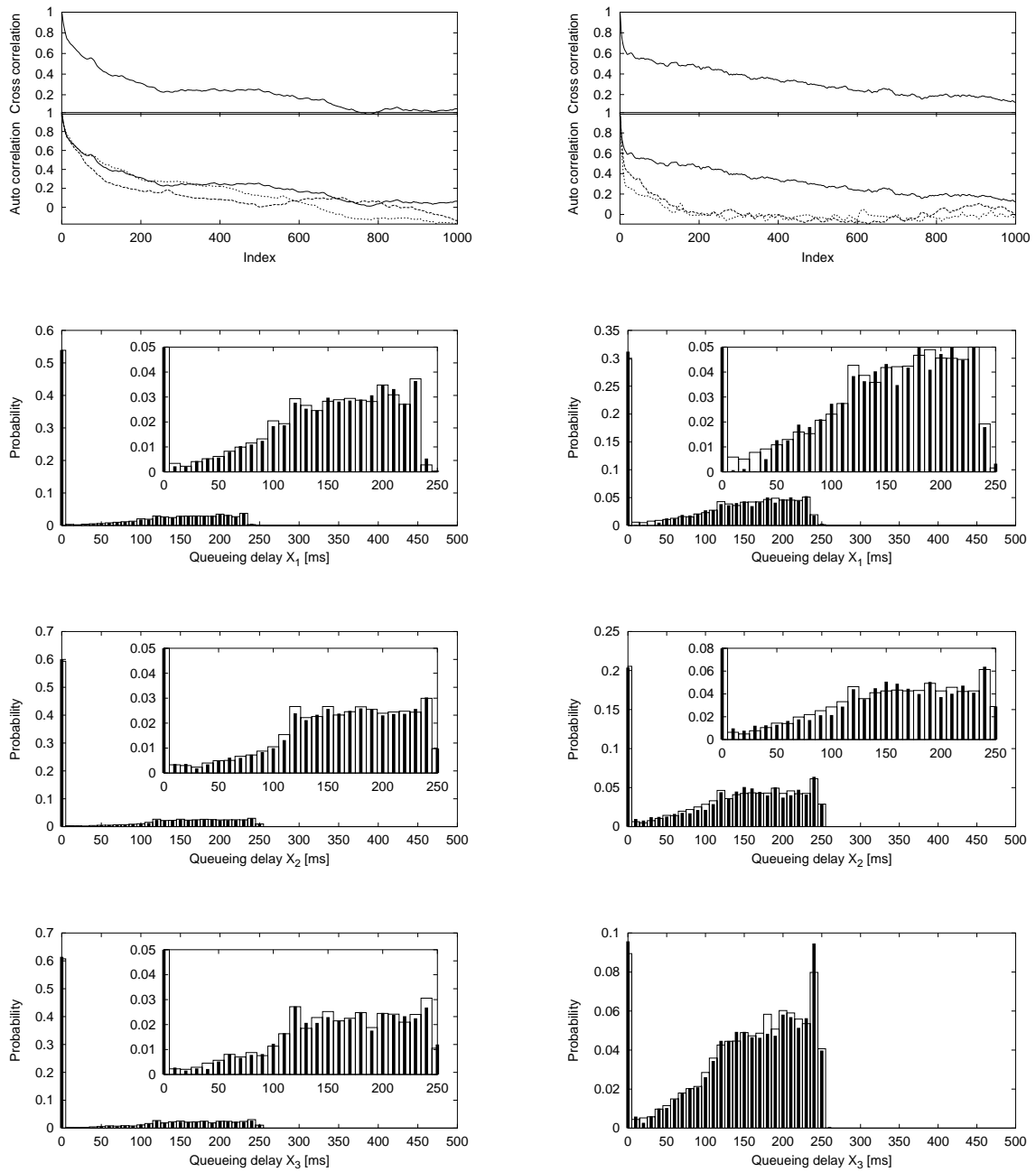
In the simulations the source node (1) sent 10000 pairs of 40 Byte UDP packets back-to-back with an inter-pair period of 0.1s, where the two components in a pair were destined to the distinct receiver nodes (6) and (7). Fluctuating background traffic was present in different proportions on the links indicated by dashed arrows. It is well known that the majority of the traffic in the current Internet is carried by the transmission control protocol (TCP), thus we modeled the background traffic with a superposition of many TCP(Reno) connections, where the initiation times and the durations of the connections were chosen respectively from a uniform and exponential distributions. We performed simulations with two different settings of background traffic intensity, mimicking cases of low and high utilization of the network. The details of the parameter settings are summarized in table 1, where we use the following notations:  $N_{TCP}$  stands for the number of TCP connections initiated

on a given link  $k \rightarrow l$ , while  $\langle ON \rangle$  gives the average ON periods of the TCP connections on that link. The output of the simulations were the logged time-series of end-to-end delays, that after minimum filtering provided the end-to-end queueing delay time-series ( $Y_2, Y_3$ ). For comparison purposes the simulations also logged the time-series of one-way delay on the different segments of the underlying two-leaf tree. We applied the inference method of Section 2.2 on the input time-series ( $Y_2, Y_3$ ) with the parameters of  $q = 10$  ms,  $B = 50$ ,  $\epsilon = 0.0001$ , and using the evenly distributed initial probabilities  $P_{i,j}^0 = 1/(B + 1)$ . Figure 6 shows the convergence of the EM algorithm, where we see that it provides good estimates already after a few tens of iterations. It is also apparent that equation (8) with the given value of  $\epsilon$  is a good indicator of convergence. The re-



**Figure 6. The convergence of the calculated value of the log-likelihood function toward the maximum, in the case of the high utilization case. The inset shows the probabilities  $P_{3,j}$ ,  $j \in (0, 1..Bq)$  after each iteration of the EM algorithm.**

sults after the termination of the EM algorithm are shown in Fig. 5. We find impressive agreement of the inferred and the real queueing delay distributions, despite the fact that the probe delays on a given segment show long-range temporal correlations, as can be seen in the lower part of the top figures in Fig. 5. It is also apparent that the interference of the probe stream with the background packets did not destroyed the high correlation on the common segment between the probe delays in the same pair. We find a correlation index of  $C = 1$  in both cases. It seems that the main effect of the background traffic was that one or both of the probes were dropped for some packet-pairs, however these pairs were discarded from the analysis. Indeed we find that out of the 10000 packet-pairs only 9681 has arrived successfully to the destinations in the low utiliza-



**Figure 5.** The correlation properties and the true vs. inferred queuing delay distributions corresponding to the ns-2 simulations of the low (left column) and high (right column) utilization cases. From top to bottom the figures indicate the cross correlation of the delay experienced by the different packets in a pair on the common segment, the auto correlation of the delays on the different segments, and the true (boxes) vs. inferred (impulses) queuing delay distributions. The insets show the magnification of the distributions.

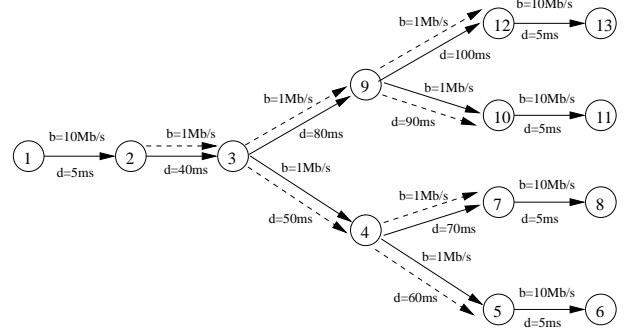
**Table 2. The inferred (left columns) and the measured (right columns) statistics of the queueing delays (in milliseconds), on the different network segments, in the low and high network utilization cases.**

$i$	$E(X_i)$	$\sqrt{E(X_i - E(X_i))^2}$	$\langle X_i \rangle$	$\sqrt{Var(X_i)}$
low				
1	72.7	98.6	71.8	85.6
2	65.9	71.7	66.8	88.1
3	63.0	69.1	63.9	87.1
high				
1	110.2	76.9	110.0	85.4
2	127.0	86.7	127.2	83.6
3	152.4	69.3	152.6	72.6

tion case, and 8946 in the high utilization case. Given the distributions of the internal queueing delays one can also determine all of its moments. The most relevant statistics in our case are the mean  $E(X_i) = \sum_{j=0}^B X_{i,j} P_{i,j}$  and the standard deviation  $\sqrt{E(X_i - E(X_i))^2}$ . Table 2 lists the results of the inferred moments that can be compared to the sample mean  $\langle X_i \rangle$ , and the square root of the sample variance  $Var(X_i)$  of the logged queueing delay time-series on the different segments. We can observe superior agreement for the mean of the distributions in both cases, where the discrepancy of the real and the inferred mean is less than the size of a single bin used in the quantization ( $q = 10$  ms). Reasonable agreement is also found for the standard deviations. Here the discrepancy between the inferred and measured values is found to be less than the bin size in the high utilization case, and being on the order of the bin size in the low utilization case. Note that for the variance estimation on the different segments there also exist a trivial approach, which involves the measurement of the sample end-to-end covariance.

### 3.1 Inference in a four receiver network

In this subsection we investigate how the performance of the inference method scales to network topologies larger than the two leaf-tree on a specific example. Figure 7 indicates the network scenario simulated in ns-2, which has the same topology as the tree depicted in Fig 1 until level 2. The background traffic on each link, indicated by dashed arrows, were modeled by the same approach as earlier in this section with the parameters  $N_{TCP} = 100$  and  $\langle ON \rangle = 20$ . For any other parameter not indicated here we used the same value as previously. To resolve all the network segments, we followed the minimal approach and sent packet-pairs in a round-robin fashion to the destination node pairs (6,8), (11,13), and (6,13). The simulation provided the end-to-end delays, that were subsequently minimum filtered to produce the end-to-end queueing delay



**Figure 7. The network scenario built in ns-2 with 4 receiver nodes.**

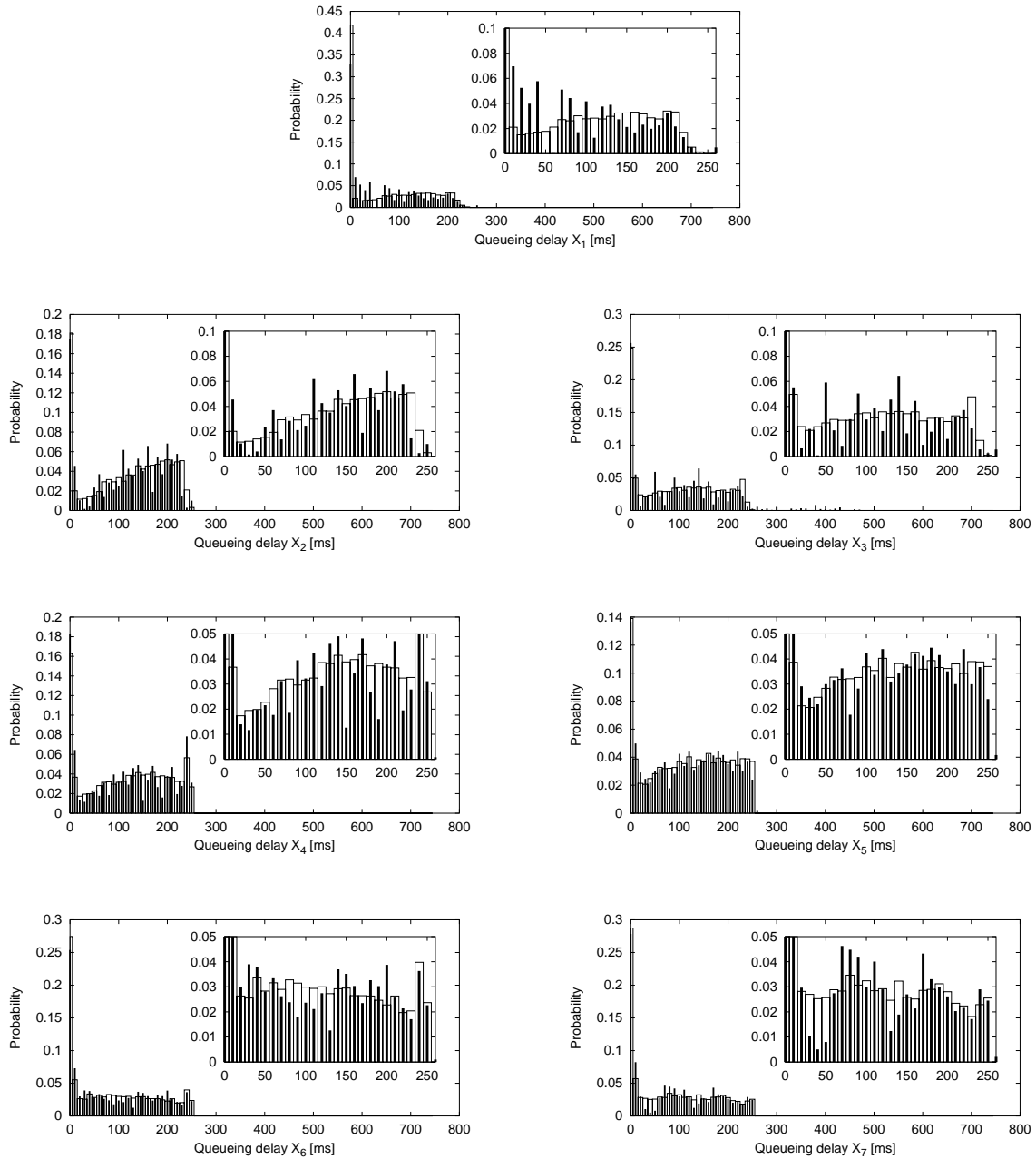
**Table 3. The inferred (left columns) and the real (right columns) statistics of the queueing delays (in milliseconds), on the different network segments, in the 4 receiver network scenario.**

$i$	$E(X_i)$	$\sqrt{E(X_i - E(X_i))^2}$	$\langle X_i \rangle$	$\sqrt{Var(X_i)}$
1	67.0	56.9	73.1	77.0
2	130.4	79.9	121.0	79.5
3	99.4	82.5	94.7	80.3
4	114.0	81.3	117.6	81.7
5	115.1	75.4	118.3	80.6
6	89.8	83.6	88.6	82.9
7	88.0	86.6	86.5	82.0

time-series  $(Y_4, Y_5)$ ,  $(Y_6, Y_7)$ , and  $(Y_4, Y_7)$ . Here we use the labeling of the network segments as described in subsection 2.1. The inference method of Section 2 was applied to these time-series with the parameters of  $q = 10$  ms,  $B = 74$ ,  $\epsilon = 0.0001$ , with uniformly distributed initial probabilities. This produced the distributions of the queueing delays  $X_1, X_4, X_5, X_6, X_7$ , and the convolutions  $Conv(X_1, X_2)$ ,  $Conv(X_1, X_3)$ , and  $Conv(X_2, X_4)$ ,  $Conv(X_3, X_7)$ . Finally the remaining distributions of the queueing delays  $X_2$  and  $X_3$  were determined by performing the numerical deconvolutions with NNLS and averaging over the different cases. The results of the distributions and of the statistical moments can be compared to the true cases in Fig. 8 and table 3. Here we can again observe that the inference method predicts very accurately the the average queueing delays, and it also provides good estimates for the standard deviation.

## 4 Network tomography measurements: test in a LAN environment

The impressive performance of the inference method in the ns-2 simulations shows the promise, that it will serve as a useful tool in real delay tomography measurements in the



**Figure 8. The true (boxes) vs. inferred (impulses) queuing delay distributions on the different segments of the measurement tree corresponding to the ns-2 simulation scenario of Fig. 7. The insets again show the magnification of the distributions.**

Internet. To perform delay tomography in a real network poses several technical challenges. First in order to be able to measure one-way delay, source and receiver nodes has to be synchronized to a common clock-reference, and must stay in the synchronized state during the measurements.

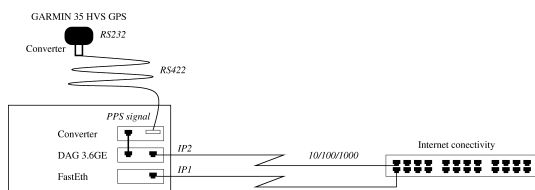
Second, the measuring infrastructure has to be very precise in order to resolve also the small queuing delay components associated to the high-bandwidth (Gigabit) links.

As a subproject of the European Union sponsored EVERGROW Integrated Project, Currently we are develop-

ing a state of the art high-precision measuring infrastructure that on the long run is planned to consist of 50 measuring nodes deployed at different locations in Europe. This platform, the Evergrow Traffic Observatory Measurement InfrastruCture (ETOMIC), among others will provide the ability to perform delay and loss tomography based on unicast probing techniques, and will be generally available and open to the public. Before such an infrastructure could begin to operate it is crucial to test the tomography tools in a real, but *controlled* environment. In order to be able to evaluate the performance of the different inference algorithms, one has to access the probe traffic not only at the end-nodes, but also at the branching nodes. In a real wide-area network (WAN) this requirement is not attainable in general, thus we have to resort to make our test measurements in a LAN environment.

#### 4.1 Description of the measurement setup

In our first test measurement we arranged 4 measuring nodes in the topology of the two-leaf tree (see Fig. 2). The schematics of a measuring node is displayed in Fig. 9. These are based on standard PC hardware, and include an Endace DAG card as the network monitoring interface, which is specifically designed for active and passive monitoring. Such cards do not use interrupts to signal packet arrivals to the kernel, thus packets can be captured at gigabit speeds. DAG cards also provide advanced capabilities for transmission, a burst composed by several packets can be transmitted with precise user-defined inter packet timings (nanoseconds). The measuring nodes are synchronized by GPS (Garmin GPS 35 HVS), that provides a PPS (pulse per second) reference signal directly to the DAG card reference signal. The resulting accuracy is 100 nanoseconds in the packet timestamps and interframe generation intervals. The measuring nodes were connected together by 100



**Figure 9. The measuring node is a standard PC equipped with DAG 3.6GE network interface card, connected to a GPS unit.**

Mb/s links with network switches placed on each path. The probe stream consisted of pairs of 68 Byte UDP packets sent back-to-back with an inter-pair period of 0.1 s. On each segment of the measurement tree we have generated Poisson distributed background traffic, consisting of 1500

Byte UDP packets, which was mixed to the probe stream through the switches situated between the nodes. The probes were generated and launched by a DAG 3.6GE card from the source node, and similar cards in the two receiver nodes were used to capture them. We also used a DAG 3.5E dual port card that monitored the probe stream at the branching node. All these measuring nodes were synchronized by GPS reference signals, thus we could measure accurately the end-to-end delays from the source to receivers, and also the one-way delays on each segment of the two-leaf tree that are needed for comparison purposes.

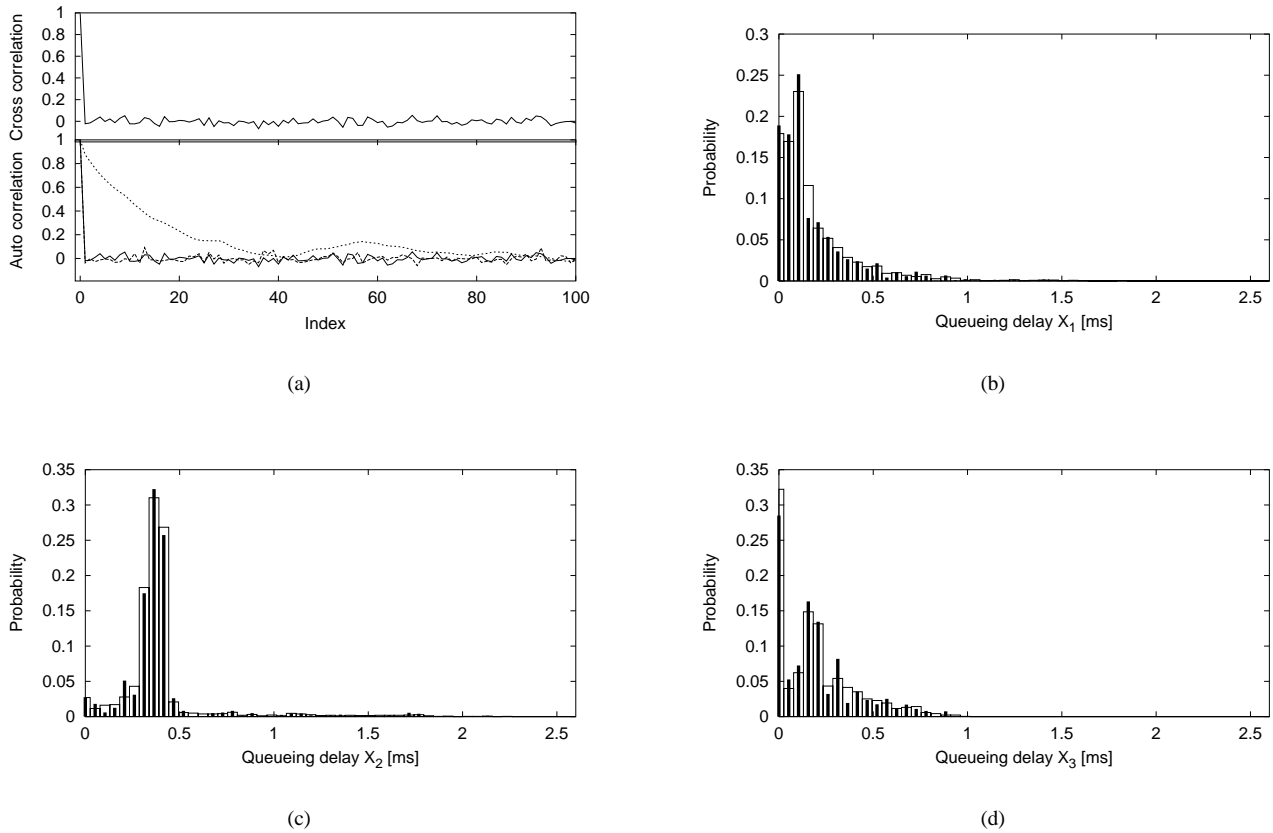
#### 4.2 Measurement results and discussion

Since the probe stream was monitored at the branching node, this allowed us to investigate the correlation properties of the probe delay on each segment of the measurement tree and thus test the validity of the assumptions that the inference technique is based on. The upper part of Fig. 10(a) shows the cross-correlation function of the delay time-series measured by the two different components in the packet-pair on the common segment from the source to the branching node. It is apparent that the delay experienced by the probes in the same pair are highly correlated (with a correlation index of  $C \approx 0.9934$ ), thus we see that using a high-precision synchronized measuring infrastructure the basic assumption of the inference method of Section 2.2 is satisfied.

The lower part of the same figure indicates the auto-correlation functions of the delay time-series measured by the probes on the three different segments. On two of the segments the delay of subsequent probes can be considered independent, however on one of the leaves we find long-range correlations as in the ns-2 simulations of Section 3 As was the case in Section 3, we can expect that despite the fact that the assumption of temporal independence of the probe delays is violated, this will not affect the quality of the inferred queuing delay distributions.

Indeed, looking at subfigures (b), (c), (d) of Fig.10 one finds a very good agreement between the inferred and the measured queuing-delay distributions. To obtain these results first the constant part of the delay were eliminated from the measured time-series by minimum filtering. The quantization parameters of the inference algorithm,  $q = 0.052$  ms, and  $B = 50$  were chosen so that the measured end-to-end queuing delays could just fit in the interval of  $(0 \leq Y_i \leq Bq)$ ,  $i \in (2, 3)$ . Here we used the evenly distributed probabilities  $P_{i,j} = 1/(B+1)$  as the initial condition to the EM algorithm, and an error limit of  $\epsilon = 0.0001$ . With these settings the EM algorithm converged after 325 iterations.

In table 3 we list the mean and standard deviation of the inferred queuing delay distributions, that can be com-



**Figure 10. Correlation properties of the measured delay time-series (a), and the comparison of the measured (boxes) and inferred (impulses) queueing delay distributions in the two-leaf tree LAN network scenario. Subfigure (b) shows the results on the common segment, while (c) and (d) for the two leaves. See the text for additional details and discussion.**

**Table 4. The inferred (left columns) and the measured (right columns) statistics of the queueing delays on the different network segments (in milliseconds).**

$i$	$E(X_i)$	$\sqrt{E(X_i - E(X_i))^2}$	$\langle X_i \rangle$	$\sqrt{Var(X_i)}$
1	0.172	0.2105	0.185	0.2129
2	0.393	0.2431	0.395	0.2453
3	0.195	0.2019	0.193	0.1986

pared to the sample mean and the square-root of the sample variance of the associated measured queueing delay time-series. Observing the values in the table we again see that the discrepancy between the measured and the inferred statistical moments is less than the value of the chosen bin size

( $q = 0.052$  ms).

## 5 Conclusion

In this paper we presented a method for estimating internal queueing delay distributions from end-to-end measurements. The method uses quantization of the delays and employs the EM algorithm to find the maximum likelihood estimate of the queueing delay probabilities. We have tested the performance of the method in realistic simulations of different network scenarios and revealed impressive agreement of the inferred and the real queueing delay distributions and of their moments.

The potential to use network tomography has also been demonstrated in a real measurement performed in a local-area scenario. It is found that using a high-precision, GPS-

synchronized infrastructure the basic assumption of the tomography approach is fulfilled, and good agreement between the real and inferred queueing delay distributions are obtained. These results facilitate network tomography as an efficient measurement technique of queueing delays in the Internet.

## 6 Acknowledgements

The authors acknowledge the support of the European Union IST FET Integrated Project EVERGROW.

## References

- [1] *Ucb/lbnl/vint network simulator - ns (version 2.27)*, <http://www.mash.cs.berkeley.edu/ns/>.
- [2] T. Bu, N. Duffield, F. L. Presti, and D. Towsley. Network tomography on general topologies. In *Proc. of ACM Sigmetrics 2002*. Marina del Rey, CA, 2002.
- [3] R. L. Carter and M. E. Crovella. Measuring bottleneck link speed in packet-switched networks. *Technical Report, BU-CS-96-006, Boston University*, December 1996.
- [4] M. Coates and R. Nowak. Network loss inference using unicast end-to-end measurement. In *Proc. ITC Conf. IP Traffic, Modeling and Management*. Monterey, CA, September 2000.
- [5] M. Coates and R. Nowak. Network tomography for internal delay estimation. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*. Salt Lake City, May 2001.
- [6] N. Duffield, J. Horowitz, F. L. Presti, and D. Towsley. *Network Delay Tomography from End-to-end Unicast Measurements*. Springer Verlag - Lecture Notes in Computer Science, Berlin, 2001.
- [7] C. L. Lawson, and B. J. Hanson. *Solving Least Squares Problems*. Prentice-Hall, Englewood Cliffs, NJ, 1974.
- [8] N. Duffield and F. L. Presti. Multicast inference of packet delay variance at interior network links. In *Proc. of IEEE Infocom 2000*. Tel Aviv, Israel, March 2000.
- [9] N. Duffield, F. L. Presti, V. Paxson, and D. Towsley. Inferring link loss using striped unicast probes. In *Proc. of IEEE Infocom 2001*. Anchorage, AK, April 2001.
- [10] G. J. McLachlan and T. Krishnan. *The EM algorithm and extensions*. John Wiley, New York, 1997.
- [11] S. Moon, P. Skelly, and D. Towsley. Estimation and removal of clock skew from network delay measurements. In *Proc. of Infocom 99*, pages 227–234. New York, March 1999.
- [12] V. Paxson. On calibrating measurements of packet transit times. In *Proc. of SIGMETRICS 98*, pages 227–234. Madison, WI, June 1998.
- [13] F. L. Presti, N. Duffield, J. Horowitz, and D. Towsley. Multicast-based inference of network-internal delay distributions. *ACM/IEEE Transactions on Networking*, December 2002.
- [14] C. J. Wu. On the convergence properties of the em algorithm. *Annals of Statistics*, 11(1):95–103, December 1983.