Quantile Sampling for Practical Delay Monitoring in Internet Backbone Networks

Baek-Young Choi [a,*], Sue Moon [b], Rene Cruz [c], Zhi-Li Zhang [d], Christophe Diot [e]

[a] *University of Missouri, Kansas City, MO, USA*

[b] *Korea Advanced Institute of Science and Technology, Daejeon, Korea*

[c] *University of California, San Diego, CA, USA*

[d] *University of Minnesota, Twin Cities, MN, USA*

[e] *Thomson Research, Paris, France*

Abstract

Point-to-point delay is an important network performance measure as it captures service degradations caused by various events. We study how to measure and report delay in a concise and meaningful way for an ISP, and how to monitor it efficiently. We analyze various measurement intervals and potential metric definitions. We find that reporting high quantiles (between 0.95 and 0.99) every 10-30 minutes as the most effective way to summarize the delay in an ISP. We then propose an active probing scheme to estimate a high quantile with bounded error. We show that only a small number of probes are sufficient to provide an accurate estimate. We validate the proposed delay monitoring technique on real data collected on the Sprint IP backbone network. To make our work complete, we lastly compare the overhead of our active probing technique with a passive sampling scheme and show that for delay measurement, active probing is more practical.

Key words: Delay, Performance monitoring, Active probing

1. Introduction

Point-to-point delay is a powerful "network health" indicator in a backbone network. It captures service degradation due to congestion, link failure, and routing anomalies. Obtaining meaningful and accurate delay information is necessary for both ISPs and their customers. Thus delay has been used as a key parameter in Service Level Agreements (SLAs) between an ISP and its customers [12, 33]. In this paper, we systematically study how to measure and report delay in a concise and meaningful way for an ISP, and how to monitor it efficiently.

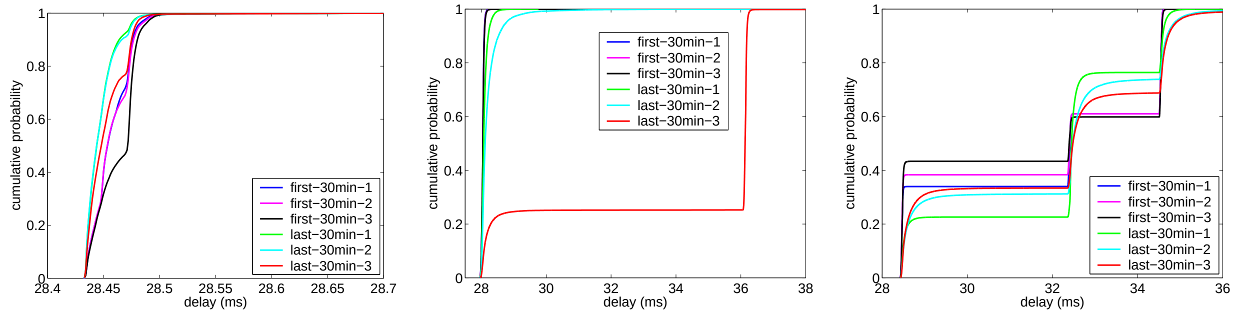
Operational experience suggests that the delay metric should report the delay experienced by most packets in the network, capture anomalous changes, and not be sensitive to statistical outliers such as packets with options and transient routing loops [3, 11]. The common practice in operational backbone networks is to use `ping`-like tools. `ping` measures network round trip times (RTTs) by sending ICMP requests to a target machine over a short period of time. However, `ping` was not designed as a delay measurement tool, but a reachability tool. Its reported delay includes uncertainties due to path asymmetry and ICMP packet generation times at routers. Furthermore, it is not clear how to

* Corresponding author. tel.:+1 816 235-2750; fax: +1 816 235 5159.

Email address: `choiby@umkc.edu` (Baek-Young Choi).

Table 1
Summary of matched traces (delay in ms)

Set	From	To	Duration	Packets	min.	Avg.	med.	$.99^{th}$	max.
1	OC-48	OC-12	16h 24m	1,349,187	28.430	28.460	28.450	28.490	85.230
2	OC-12	OC-12	5h 27m	882,768	27.945	29.610	28.065	36.200	128.530
3	OC-12	OC-48	5h 21m	3,649,049	28.425	31.805	32.425	34.895	135.085



(a) Data Set # 1 (b) Data Set # 2 (c) Data Set # 3

Fig. 1. Empirical cumulative probability density function of delay over 30 minute interval

set the parameters of measurement tools (e.g., the test packet interval and frequency) in order to get a certain accuracy.

Inaccurate measurement defeats the purpose of performance monitoring. In addition, injecting a significant number of test packets for measurement may affect the performance of regular traffic, as well as tax the measurement systems with unnecessary processing burdens. More fundamentally, defining a metric that can give a meaningful and accurate summary of point-to-point delay performance has not been considered carefully.

We raise the following practical concerns in monitoring delays in a backbone network. How often should delay statistics be measured? What metric(s) capture the network delay performance in a meaningful manner? How do we implement these metrics with limited impact on network performance? In essence, we want to design a practical delay monitoring tool that is amenable to implementation and deployment in high-speed routers in a large network, and that reports useful information.

The major contributions of this paper are threefold: (i) By analyzing the delay measurement data from an operational network (Sprint US backbone network), we identify high-quantiles [0.95-0.99] as the most meaningful delay metrics that best reflect the delay experienced by most of packets in an operational network, and suggest 10-30 minute time scale as an appropriate interval for estimating the high-quantile delay metrics. The high-quantile delay metrics estimated over such a time interval provide a best representative picture of the network delay performance that captures the major changes and trends, while they are less sensitive to transient events, and outliers. (ii) We propose and develop an active probing method for estimating high-quantile delay metrics. The novel feature of our proposed method is that it uses the minimum number of samples needed to bound the error of quantile estimation within a prescribed accuracy, thereby reducing the measurement overheads of active probing. (iii) We compare the network wide overhead of active probing and passive sampling for delays. To the best of our knowledge, this is the first effort to propose a complete methodology to measure delay in operational networks and validate the performance of the active monitoring scheme on operational data.

The remainder of this paper is organized as follows. In Section 2 we provide the background and data used in our study. In Section 3 we investigate the characteristics of point-to-point delay distributions obtained from the packet traces and discuss metrics used in monitoring delay in a tier-1 network. In Section 4 we analyze how sampling errors can be bounded within pre-specified accuracy parameters in high quantile estimation. The proposed delay measurement scheme is presented and its performance is evaluated using packet traces in Section 5. In Section 7 we summarize related works. We conclude the paper in Section 8.

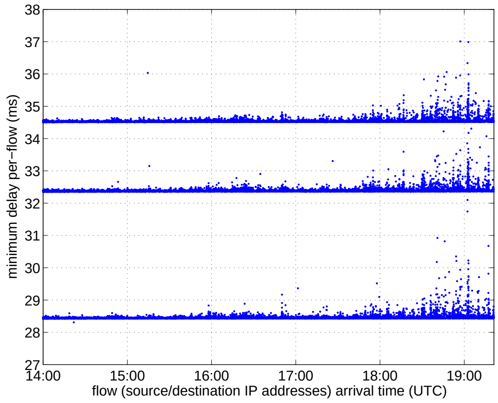


Fig. 2. Presence of ECMP in Data Set 3

2. Data and Background

We describe our data set and provide some background about point-to-point delay observed from this data.

2.1. *Data*

We have collected packet traces from Sprint's tier-1 backbone using the methodology described in [9]. The monitoring system passively taps the fibers to capture the first 44 bytes of all IP packets. Each packet header is timestamped. The packet traces are collected, from *multiple* measurement points *simultaneously*, and span over a *long period* of time (e.g. hours). All the monitoring systems are synchronized by GPS (Global Positioning System). The resolution of the clock is sub-microsecond, allowing us to disambiguate packet arrival times on OC-48 links. The timestamp maximum error is 5 microseconds.

To obtain packet delays between two points, we first identify packets that traverse two points of measurements. We call this operation *packet matching*. We use hashing to efficiently match two packet traces. We use 30 bytes out of the first 44 bytes in the hash function. The other 14 bytes are IP header fields that would not help disambiguate similar packets (e.g. version, TTL, and ToS). We occasionally find duplicate packets. Since these packets are totally identical, they are a source of error in the matching process. Given that we observe less than 0.05% of duplicate packets in all traces, we remove these duplicate packets from our traces.

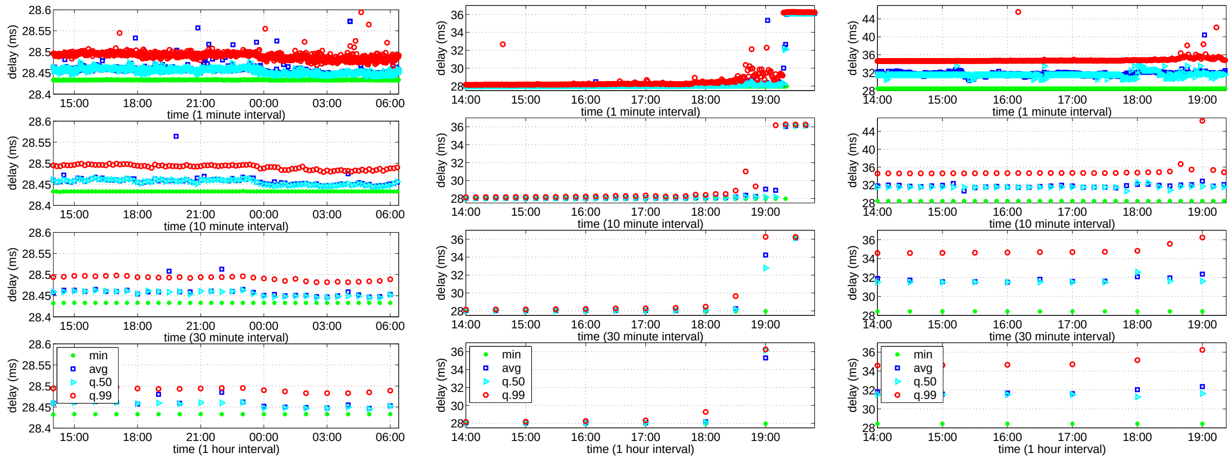
We have matched more than 100 packets traces, and kept only those *matched trace* that exhibited many (more than half a million) successful matched packets. The matched traces are from paths with various capacities and loads over multihop nodes. For a succinct presentation, we have chosen to illustrate our observations of with 3 matched traces out of the 21 we studied. The traces shown are representative and the other traces show similar results. The statistics of these three matched trace are shown in Table 1. In all the matched trace data sets, the source and destination links are located on the West Coast and the East Coast of the United States respectively, rendering trans-continental delays over multiple hops.

2.2. *Background*

We now briefly discuss the characteristics of actual packet delays observed on the Sprint US IP backbone. More detailed observations can be found in [25, 4].

The empirical cumulative probability distributions of point-to-point delays using a bin size of 5 μs is shown Figure 1. For ease of observation, we divide the duration of traces into 30 minute intervals and then plot distributions for the first and last three intervals of each trace duration.

Delay distributions exhibit different shapes, as well as change over time, especially in Data Set #2 and #3. We explain these differences as follows. In theory, the packet delay consists of three components: propagation delay, transmission delay and queueing delay. Propagation delay is determined by the physical characteristics of the path. Transmission delay is a function of the link capacities along the path as well as the packet size. Queueing delay depends on the traffic load along the path, and thus varies over time. In practice, other factors add variations to the delay packets experience in an operational network. First, Internet packet sizes are known to exhibit three modes, where the peaks are around 40, 576 (or 570), and 1500 bytes [14]. When there is little queueing on the path, the packet size may impact the shape of a distribution even in the multi-hop delays, as shown in Figure 1(a). In addition, routing can introduce delay variability. Route may change over time because of link failure. Figure 1(b) shows that the path between the two measurement points changed within the last 30 minutes. Furthermore, packets can take multiple routes between two points because of load balancing, as in Figure 1(c). Equal-cost multi-path (ECMP) routing [34] is commonly employed in operational net-



(a) Data Set #1 (b) Data Set #2 (c) Data Set #3

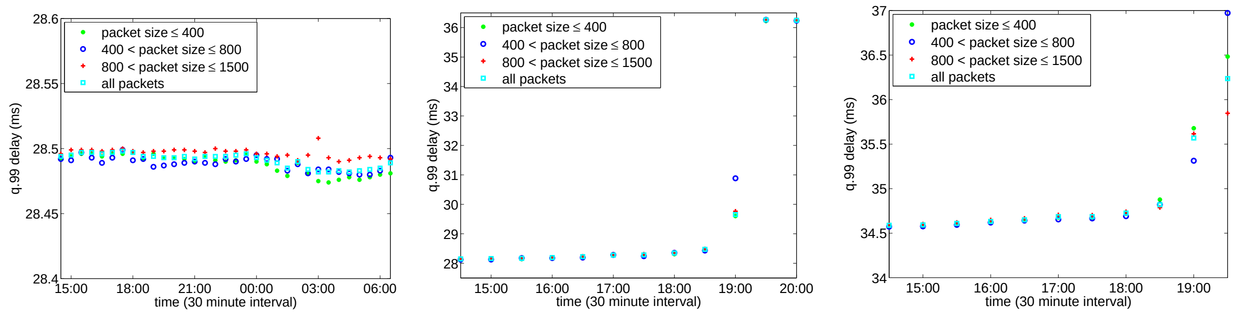
Fig. 3. Delay metrics over different estimation intervals

works. Routers (e.g., Cisco routers in our study) randomly split traffic using a hash function that takes the source and the destination IP addresses, and the router ID (for traffic splitting decision to be independent from upstream routers) as input to determine the outgoing link for each packet. Therefore packets with the same source and destination IP addresses always follow the same path. We define a *(two-tuple) flow* to be a set of packets with the same source and destination IP addresses, and group packets into flows. We then compute the *minimum* packet delay for each flow. As suggested in [4], if the two flows differ significantly in their minimum delays, they are likely to follow two different paths. In Figure 2 we plot the minimum delay of each flow by the arrival time of the first packet in the flow for Data Set 3. The plot demonstrates the presence of three different paths, each corresponding to one step in the cumulative delay distribution of Figure 1(c). Last, extreme packet delays may occur even under a perfectly engineered network, due to routing loops [11] or router architecture [3] related issues. From the perspective of a practical delay monitoring, we need to take all these factors into account to provide an accurate and meaningful picture of actual network delay.

3. Metrics Definition for Practical Delay Monitoring

The objective of our study is to design a practical delay monitoring tool to provide a network operator with a *meaningful* and *representative* picture of delay performance of an operational network. Such a meaningful and representative picture should tell the network operator *major* and *persistent* changes in delay performance (e.g., due to persistent increase in traffic loads) *not* transient fluctuations due to minor events (e.g., a transient network congestion). Hence in designing a practical delay monitoring tool, we need to first answer two inter-related questions: (i) what metrics should we select so as to best capture and summarize the delay performance of a network, namely, by a majority of packets; and (ii) over what time interval should such metrics be estimated and reported? We refer to this time interval as the (metrics) *estimation* interval. Such questions have been studied extensively in statistics and performance evaluation (see [17], for a general discussion of metrics in performance evaluation). From the standpoint of delay monitoring in an operational network, we face some unique difficulties and challenges. Thus our contribution in this respect lies in putting forth a practical guideline through detailed analysis of delay measurements obtained from Sprint's operational backbone network: we suggest *high quantiles ([0.95,0.99]) estimated over a 10-30 minute time interval* as meaningful metrics for ISP practical delay monitoring. In the following we present our analysis and reasoning using the three data sets discussed in the previous section as examples.

To analyze what metrics provide a meaningful and representative measure of network delay performance, we consider several standard metrics, i.e., minimum, average, maximum, median (50% percentile, or 0.5th quantile) and high quantiles (e.g., 0.95th quantile), estimated over various time inter-



(a) Data Set #1 (b) Data Set #2 (c) Data Set #3

Fig. 4. Impact of packet size on quantile (30 minute estimation interval)

vals (e.g., 30 seconds, 1 minute, 10 minutes, 30 minutes, 1 hour), using the delay measurement data sets collected from the Sprint operational backbone networks. Results are plotted in Figure 3. Note that here we do not plot the maximum delay metrics as maximum delays are frequently so large that they obscure the plots for the other metrics. Some statistics of the maximum delays are given in Table 1, where we see that maximum delays can be several multiples of the 0.99^{th} quantiles.

From the figures, we see that delay metrics estimated over small time intervals (e.g., 1-minute) tend to fluctuate frequently, and they do not reflect significant and persistent changes in performance or trends (for example, Figure 3(a), Figure 3(b) at time 14:40 and Figure 3(c) at time 16:30). [1]

On the other hand, the increase in delay around 18:30 and onwards in both Data Set #2 and Data Set #3, represents a more significant change in the delay trend, and should be brought to the attention of network operators. Note also that in a few occasions the average delays particularly *estimated over a small time interval* are even much larger than the 0.99^{th} quantiles (see, the top two plots in Figure 3(a) around 18:00 and 21:00) – this is due to the extreme values of the maximum delays that drastically impact the average.

As a general rule of thumb, the time interval used to estimate delay metrics should be large enough not to report transient fluctuations, but not too large in order to capture in a timely fashion the major changes and persistent trends in delay performance. In this regard, our analysis of the data sets suggests that 10-30 minute time interval appear to be an appropriate delay estimation interval. As an aside, we remark that our choice of 10-30 minute time interval is also consistent with the studies of others using different measurement data. For example, the active measurement study in [36] using NIMI measurement infrastructure [28] has observed that in general packet delay on the Internet appears to be steady on time scales of 10-30 minutes.

In choosing delay metrics, similar properties are desired. A meaningful metric to ISPs should characterize the delay experienced by most of packets, thereby providing a good measure of the typical network performance experienced by network users. Furthermore, such a delay metric should not be too sensitive to outliers. We summarize the pros and cons of various delay metrics as below:

– Maximum delay suffers greatly from outliers. Some packets might experience extreme delays even under well-managed and well-provisioned networks [11, 13, 20] due to IP packets with options, malformed packets, router anomalies and transient routing loop during a convergence time. The rate of outliers is such that there would be such a packet in almost every time interval. However, packets that experience the maximum delay are not representative of the network performance.
– Average or median delay have the main disadvantage of not capturing delay variations due to route changes (Figure 1(b)) or load-balancing (Figure 1(c)) that happen frequently in operational networks. Moreover, average is sensitive to outliers especially when a small number of test packets are used.
– Minimum delay is another commonly used metrics. We can see from Figure 3 that the minimum delay is very stable at any time granularity. A change in minimum delay reports a change in the

[1] We do not know exactly what caused the delays. We focus our work on *measuring* and *estimating* delays. and investigating reasons of the delay is out of the scope in our work.

shortest path.

– High quantiles ([0.95, 0.99]) ignore the tail end of the distribution and provides a practical upper bound of delay experienced by most of the packets. When estimated over the appropriate time interval, it is not sensitive to a small number of outliers. However, in the presence of multiple paths between the measurement points, high quantiles reflects only the delay performance of the longest path.

Weighing in the pros and cons of these metrics, we conclude that high percentile is the most meaningful delay metric. However, high quantile does not detect a change in the shortest path. Together with minimum delay, it gives an ISP the range of delays experienced by most of the packets between the two endpoints. As minimum delay is easy to capture [18] using active test packets, in this paper, we focus on the accurate estimation of high quantiles.

4. Quantile Estimation Analysis

In this section we develop an efficient and novel method for estimating high-quantile delay metrics: it estimates the high-quantile delay metrics within a prescribed error bound using a number of required test packets. In other words, it attempts to minimize the overheads of active probing. In the following, we first formulate the quantile estimation problem and derive the relationship between the number of samples and the estimation accuracy. Then, we discuss the parameters involved to compute the required number of samples.

We derive the required number of test packets to obtain a pre-specified accuracy in the estimation using Poisson modulated probing. Active test packets perform like passive samples under the following two assumptions. First, the amount of test packets should be negligible compared to the total traffic, so that it does not perturb the performance it measures. Second, the performance of test packets should well represent the performance of regular traffic. Both assumptions are held, which rationalizes our use of active probing. As we will see later, the required number of test packets is relatively small, thus it is negligible on today's high speed backbone networks. Also, we encapsulate the test packets in regular UDP packets so that they do not receive special treatments in a router, unlike packets with IP option or ICMP packets that go to the slow-path of a router.

Now, we formally define a quantile of a delay distribution. Let X be a random variable of delay. We would like to estimate a delay value q_p such that the 99%($i.e., p = 0.99$) of time, X takes on a value smaller than q_p. The value q_p is called the p^{th} quantile of delay and is the value of interest to be estimated. It is formally stated as [2]:

$$q_p = \inf\{q : F(q) \geq p\} \tag{1}$$

where $F(\cdot)$ denotes a cumulative probability density function of delay X.

Suppose we take n random samples, $X_1, X_2, \ldots, X_n$. We define $\hat{F}$, an empirical cumulative distribution function of delay, from n samples ($i = 1, \ldots, n$) as

$$\hat{F}(q_p) = \frac{1}{n}\sum_{i=1}^{n} I_{X_i \leq q_p} \tag{2}$$

where the indicator function $I_{X \leq q_p}$ is defined as

$$I_{X_i \leq q_p} = \begin{cases} 1 \text{ if } X_i \leq q_p, \\ 0 \ \text{ otherwise.} \end{cases} \tag{3}$$

Then, the p^{th} sample quantile is determined by

$$\hat{q}_p = \hat{F}^{-1}(p) \tag{4}$$

Since $\hat{F}(x)$ is discrete, $\hat{q}_p$ is defined using order statistics. Let $X_{(i)}$ be the ith order statistic of the samples, so that $X_{(1)} \leq X_{(2)} \leq \cdots \leq X_{(n)}$. The natural estimator for q_p is the p^{th} sample quantile ($\hat{q}_p$). Then, $\hat{q}_p$ is computed by

$$\hat{q}_p = X_{(\lceil np \rceil)} \tag{5}$$

Our objective is to bound the error of the p^{th} quantile estimate, $\hat{q}_p$. More specifically, we want the absolute error in the estimation $|\hat{q}_p - q_p|$ to be bounded by ε with high probability of $1 - \eta$:

$$Pr\left\{|\hat{q}_p - q_p| > \varepsilon\right\} \leq \eta \tag{6}$$

Now we discuss how many samples are required to guarantee the pre-specified accuracy using random sampling. Since they are obtained by random sampling, $X_1, X_2, \ldots, X_n$ are i.i.d. (independent and identically distributed) samples of the random variable X. It is known that quantile estimates from

[2] Note that theoretically, the original delay distribution can be considered as a continuous function, and the measured delay distribution is a realization of it.

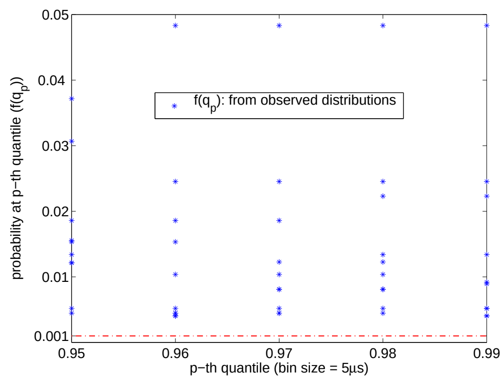


Fig. 5. Empirical tail probability

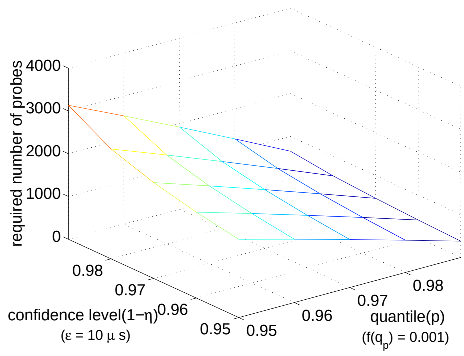


Fig. 6. Number of test packets required ($\varepsilon = 10\mu s$, $f(q_p) = 0.001$)

random samples asymptotically follow a normal distribution as the sample size increases (See Appendix for details).

$$\hat{q}_p \xrightarrow{D} N\left(q_p, \frac{\sigma^2}{n}\right) \text{ where } \sigma = \frac{\sqrt{p(1-p)}}{f(q_p)} \tag{7}$$

$f(q_p)$ is the probability density at the p^{th} quantile of the actual distribution. Eq. (7) is called Bahadur expression [31]. The estimator is known to have the following properties: (i) *unbiasedness*: the expectation of the estimate is equal to the true value (i.e., $E(\hat{q}_p) = q_p$). (ii) *consistency*: As the number of test packets n increases, the estimate converges to the true value (i.e., $\hat{q}_p \to q_p$ as $n \to \infty$). Note that the above analysis is based on *random* sampling. Thus the analysis of accuracy such as confidence interval (ε) and confidence level $(1-\eta)$ is applicable *regardless* of the underlying delay distribution from the Central Limit Theorem.

We derive from Eq. (6) and (7) the required number of samples to bound the estimation error within the pre-specified accuracy as

$$n^* = \left\lceil z_p \cdot \frac{p(1-p)}{f^2(q_p)} \right\rceil \tag{8}$$

where z_p is a constant defined by the error bound parameters (i.e., $z_p = \left(\frac{\Phi^{-1}(1-\eta/2)}{\varepsilon}\right)^2$), and $\Phi(\cdot)$ is the cumulative probability function of standard normal distribution.

Eq. (8) concisely captures the relationship of the number of samples on the quantile of interest (p), the accuracy parameters (ε, η) and a parameter of original delay distribution $(f(q_p))$.

From Eq. (7) and (8), we show that the variance of the estimate is bounded as

$$Var(\hat{q}_p) = \frac{p(1-p)}{f^2(q_p) \cdot n^*} \leq \frac{1}{z_p} \tag{9}$$

since $n^* \geq z_p \cdot \left(\frac{p(1-p)}{f^2(q_p)}\right)$.

The derivation here is for cases with low sample fractions from a large population. We have analyzed the results as if we sampled with replacement though the actual sampling is done without replacement, as it makes the formula simple and enables us to compute the required number of samples concisely. When the sampling fraction is non-negligible, an extra factor should be considered in computing the number of samples. The impact is that the actual variance from the sampling without replacement would be smaller than the one from with replacement. Thus, the actual estimation accuracy achieved is higher with the given number of samples. Practically the analysis of sampling with replacement is used as long as the population is at least 10 times as big as the sample [22].

Unfortunately, $f(q_p)$ is not known in practice. Therefore, it can only be approximated. The required number of samples is inversely proportional to $f^2(q_p)$.

A reasonable lower-bound of the value should be used in the computation of n^*, so that the accuracy of the quantile can be guaranteed. We investigate an empirical values of $f(q_p)$ using our data. The empirical p.d.f. of a delay distribution should be evaluated in terms of a time granularity of measurements. As the bin size or the time granularity of distribution gets larger, the relative frequency of delay becomes larger. In order to approximate $f^2(q_p)$, we observe the tail probabilities of delay distributions from the traces. However, for 10-30 minute durations of various matched traces from differing monitoring loca-

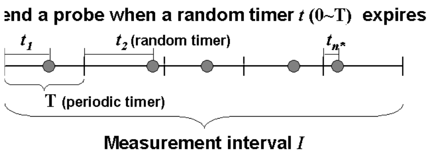


Fig. 7. Scheduling n^* pseudo-random samples

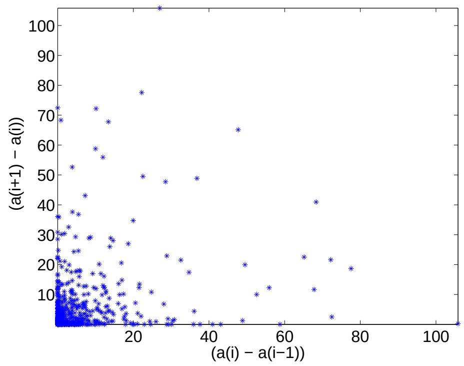


Fig. 8. Correlation of inter-packet time of long delayed packets (correlation coefficient $= 1.8e-6$)

tions and link speeds , we find that the probabilities at high quantiles, $f(q_p)$, $(0.95 \leq p \leq 0.99)$ vary little and can be reasonably lower bounded. Figure 5 shows the probability of high quantiles of the matched traces at time granularity of $5\mu s$. We find the values between 0.0005 to 0.001 are sufficient as the lower-bound of the tail probability for quantiles of $0.95 \leq p \leq 0.99$. Meanwhile, if p approaches to 1 (e.g.,$p = 0.99999$), the quantile is close to the maximum and $f(q_p)$ becomes too small requiring large number of samples. Note that when the tail probability becomes heavier, $f(q_p)$ becomes larger making the estimate more accurate. On the other hand, when the tail probability becomes smaller than the approximated, the accuracy of an estimate (the variance of estimation) would not degrade much, since the variance of the original packet delay would be small. Therefore, with given accuracy parameters and the lower bound of $f(q_p)$, the number of test packets is decided as a constant.

Figure 6 shows the number of required samples for different quantiles and different accuracy parameters [3] . It illustrates the degree of accuracy achieved with the number of samples, and thus provides a guideline on how to choose the probing frequency for a given quantile p to be estimated. A sample size between a few hundred and a few thousand test packets $(420 \sim 3200)$ is enough for $(\varepsilon = 10\mu s, 1-\eta \in [0.95, 0.99])$ range of accuracy and $(q_{.95} \sim q_{.99})$ high quantile. With high speed links (1 Gbps and above), we consider the amount of injected traffic for probing purpose negligible compared to the total traffic. For example, 1800 packets over a 10 minute period corresponds to about 3 packet per second on average. Suppose 64 byte packets are used for the test packets. This would constitute only 1.5 Kbps which is 0.0002% of the total traffic for a 30% loaded OC-48 link.

Before leaving this section we comment on estimating an entire distribution, even though our focus in this paper is on a point estimation of a most representative delay metric. Note that Eq. (8) applies to any quantile in a distribution. Thus, the estimated quantiles enjoy the pre-scribed accuracy, if the minimum required number of samples for the quantile, n^* is smaller than the used number of samples. In particular, as the quantile goes closer to median ($q = 0.5$) and the probability density at the quantile $f(q_p)$ gets larger, the required number of samples becomes smaller, resulting that the accuracy of an estimation for the quantile becomes higher.

5. Delay Monitoring Methodology

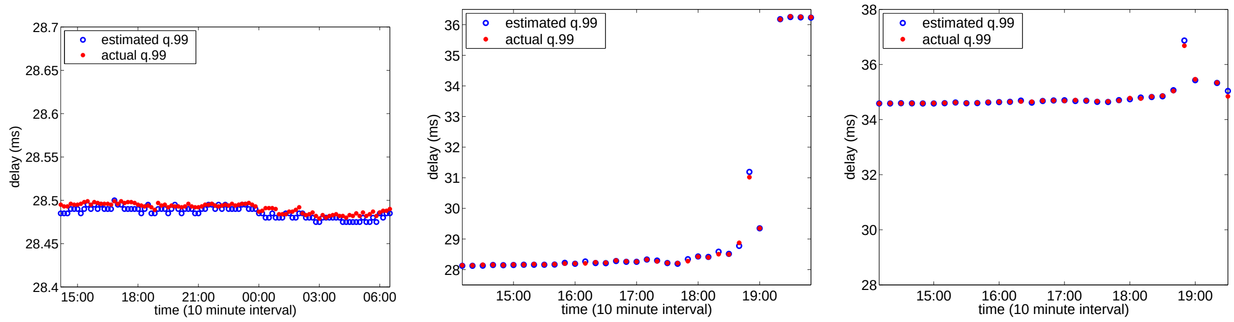
In this section, we describe our probing scheme and validate its accuracy using delay measurement data collected from the Sprint operational backbone network.

5.1. *Active Probing Methodology*

The design goal of our active probing scheme is to estimate high quantile effectively and efficiently over a fixed estimation interval. In Section 4 we have shown that at least n^* number of independent random samples are needed in the estimation interval in order to accurately estimate high quantiles.

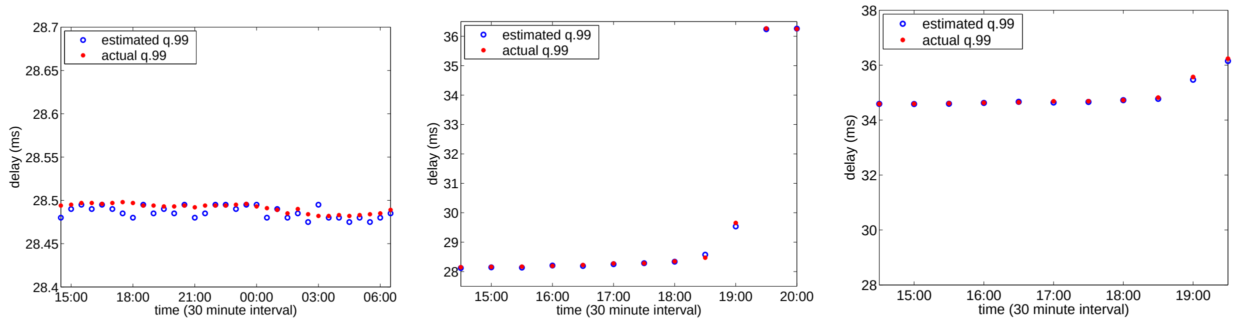
We proceed as follows. To generate n^* number of test packets within an estimation interval I, we divide the interval into n^* subintervals of length $T(=$

[3] Note that $f(q_p)$ for each high quantile is fixed equally as 0.001 from empirical observation shown in Fig. 5.



(a) Data Set #1 (b) Data Set #2 (c) Data Set #3

Fig. 9. Actual and estimated $.99^{th}$ quantiles (10 minute estimation interval)



(a) Data Set #1 (b) Data Set #2 (c) Data Set #3

Fig. 10. Actual and estimated $.99^{th}$ quantiles (30 minute estimation interval)

I/n^*). With the help of two timers – a periodic (T) timer and a random ($t \in [0, T]$) one, a random test packet is generated for each subinterval T in a time-triggered manner (i.e., whenever a random timer t expires, a test packet is generated). At the end of an estimation interval (I), the delay quantile of the test packets is computed and reported. Figure 7 illustrates graphically how to generate the pseudo-random test packets. With this scheme, we ensure that n^* number of test packets are generated independently in every estimation interval without generating a burst at any moment.

We now verify if our time-triggered *pseudo*-random probing performs close to random sampling in estimating high delay quantile. If the inter-arrival times of packets with long delays (e.g., 0.95^{th} quantile or larger) are temporarily correlated, the pseudo-random probing would not enable us to estimate high percentile delay well. However, we find that the correlation coefficient is close to 0 (for other intervals and traces with the estimation interval of 10-30 minutes). If the arrival times of packets with long delays (e.g., $.95^{th}$ quantile or larger) are temporally correlated, the pseudo-random probing may not capture the delay behavior well. Figure 8 shows the scatter plot of inter-arrival times of packets with long delays (for the last 30 minutes of Data Set #3). It illustrates that inter-arrival times of packets with long delays are essentially independent.

Test packets scheduling aside, there are several practical issues in implementing a probing scheme such as protocol type and packet size. For the type of test packets, we choose to use UDP packets instead of ICMP packets that are used in `ping`-like active probing softwares. ICMP packets are known to be handled with a lower priority at a router processor. Thus their delay may not be representative of actual packet delay. Test packet size might affect the measure of the delay. We analyzed all matched traces and found that packet size has little impact on high quantile. This is best illustrated in Figure 4 where we classify packets into three clusters based on the packet sizes, and computed their $.99^{th}$ quantile, compared with that of all packets. As observed, high quantiles from individual packet size classes are similar, and one particular packet size class does not reflect high quantile from all packets better consistently. It provides the evidence that high quantile delays are not likely to come from packets of a large size, thus the size of test packet should not impact the accuracy of high quantile estimation.

We also have performed a thorough analysis of packet properties in order to detect a correlation between packet fields and delay, if any. However, we did not find any correlation between packet types and the delay distribution. This result confirms that the tail of distribution comes from queueing delay rather than due to a special packet treatment at routers.

As ECMP is commonly employed in ISPs, we need to make sure that our test packets take all available paths when they exist. Load balancing is done on a flow basis, in order to preserve packet sequence in a flow. Therefore, we propose to vary the source address of test packets within a reasonable range (e.g., a router has a set of legitimate IP addresses for its interfaces) to increase the chances of our test packets to take all available paths. The original source address can be recorded in the test payload to allow the destination to identify the source of the test packets.

We have described the proposed active probing methodology in terms of probing schedule, the number of test packets for a certain accuracy, the test packet type and the packet size. With regard to a control protocol to initiate and to maintain monitoring sessions between endpoints, the existing protocols such as Cisco SAA (Service Assurance Agent) [30] [4] or IPPM one-way active measurement protocol (OWAMP) [24] can be used with little modification.

5.2. *Validation*

To validate the proposed technique, we emulate active test packets in the following manner [5]. Given an estimation interval (I) and accuracy parameters ($\{\varepsilon, \eta\}$), whenever the random timer (t) expires, we choose the next arriving packet from the data sets, and use its delay as an active test packet measurement. The accuracy parameters are set to be $\varepsilon = 10\mu s$ [6] and $\eta = 0.05$ to estimate $.99^{th}$ quantile of delay. We have used 0.001 and 0.0005 for $f(q_p)$. The computed numbers of samples to ensure the estimation accuracy are only 423 and 1526, respectively.

Table 2
Bounded variance of estimates ($\{\varepsilon, \eta\} = \{10\mu s, 0.05\}, p = 0.99$)

$1/z_p$	Data Set	1	2	3
25.95	$Var(\hat{q}_p)$	11.97	25.72	25.55

The estimated $.99^{th}$ quantiles over 10 minute intervals using 423 packets are compared with the actual $.99^{th}$ quantiles in Figure 9. Using the same number of 423 test packets, the estimated quantiles are compared with the actual ones over 30 minute interval in Figure 10. Using such small numbers of packets, the estimated quantiles are very close to the actual ones, for all the data sets and estimation intervals.

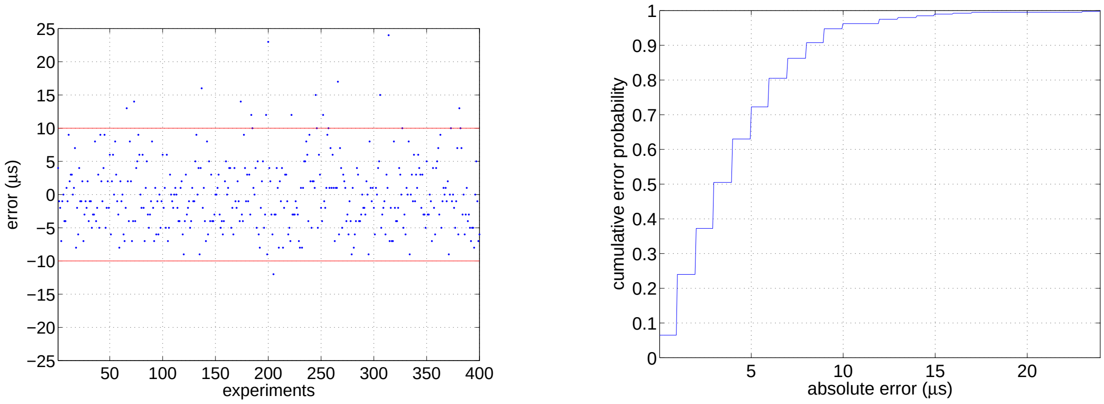
To assess the statistical accuracy, we conduct experiments over an estimation interval (30 minutes) as many as 500 times. For 0.99^{th} quantile ($q_{.99}$), we desire the error to be less than ε with probability of $1-\eta$. We compare the estimated quantile from each experiment with the actual quantile from the total passive measurements. Figure 11(a) displays the estimation error in each experiment. Most errors are less then $10\mu s$ which is the error bound ε. To validate the statistical guarantee of accuracy, in Figure 11(b), we plot the cumulative empirical probability of errors in quantile estimation. The y axis is the experimental cumulative probability that the estimate error is less than x. It illustrates that indeed 95% of the experiments give estimation error of less than $10\mu s$, which conforms to the pre-specified accuracy parameters.

Another key metric for the performance of a sampling technique is the variance of an estimator. Small variance in estimation is a desired feature for any sampling method, as it tells the estimate is more reliable. In the previous section, we have shown that the proposed scheme enables us to bound the variance of the estimates in terms of the accuracy parameters, i.e. $1/z_p = \left(\frac{\varepsilon}{\Phi^{-1}(1-\eta/2)}\right)^2$. Table 2 shows the variance of the estimates from the proposed scheme. The variances are indeed bounded by the value given in Eq. (9) given in Section 4.

[4] SAA (Service Assurance Agent) is an active probing facility implemented in Cisco routers to enable network performance measurement.

[5] We could not perform probing simultaneously to passive trace collection since all long-haul links on the Sprint backbone have been upgraded to OC-192 after the trace collection.

[6] This small error bound is chosen to show the feasibility of the proposed sampling.



(a) Absolute error of estimates

(b) Cumulative error probability

Fig. 11. Quantile estimation with bounded error ($\{\varepsilon, \eta\} = \{10\mu s, 0.05\}, p = 0.99$)

6. Active vs. Passive Sampling: Overhead Comparison

In this section, we compare a network-wide overhead of our active measurement method to a passive sampling technique.

For the comparison, we first describe a passive sampling process for delay measurement in a network (See Figure 12 for reference). We sketch here a hash based scheme proposed in [6]. For delay measurement, all regular packets are hashed and passively sampled based on their hash values and timestamped at the measurement points. To capture the same sets of packets on different measurement points, the same hash function is used to sample packets at all measurement points. Then, the collected packets are exported to a central server where the same pair of packets are identified and the delay is computed. In order to reduce the bandwidth consumed when exporting those samples, only a hash of the packet ID is exported, rather than the whole packet header. The downside of this technique is to increase the risk of packet mismatch at a central sever. The central server then matches all packets and computes the delay from the difference of their timestamps. The method can be optimized using routing information in order to ease the task of finding pairs of measurement points where packet might have traversed from one to the other.

Note that even with passive measurement, measured packets should be transferred to a central server to combine time information from measurement points, since for one delay value, *two* measurement points are involved, i.e., the source and the sink. Therefore, either active or passive, delay

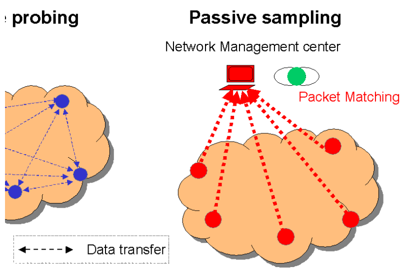


Fig. 12. Active and passive delay measurements

measurement consumes bandwidth by nature.

In order to compare the overhead, we consider the case where the number of delay samples are equal so as to achieve the same accuracy of estimation from both methods. Assuming the number of samples is small as shown in Section 4, the performance of regular traffic would not be affected by measurements for both active and passive measurement.

We ignore the control protocol overhead for signaling among routers (active probing) or between routers and a central server (passive sampling and active probing [7]), which we expect to be similar in both methods. In addition, both active and passive monitoring systems can be either implemented as an integral part of routers in an embedded manner [30] or as a stand-alone out-of-router measurement system.

First, let us analyze the bandwidth used by measurement data. Consider the number of bytes used to report one packet delay. In a passive method, we transfer only the packet identifier (hash value) rather than the entire packet header and payload.

[7] This additional signaling is required in active probing to report the estimated quantiles to a central server.

Each packet hash value should be transferred with its *timestamp*. Then, in order to compute one delay from one point to the other, two packets are required with a passive sampling. Suppose n^* packets are required for a given accuracy. For a given pair of measurement points, the number of packets that have to be sent is n^* with the active method. For the passive method, note that only a *portion* of the packets retrieved at a source router are sent to the sink router of the measurement interest. Similarly, only a portion, so called traffic *fanout factor* P, (where $0 \leq P \leq 1$) of the packets at the sink router is originated from the source router. Therefore, in order to produce the needed number of delay samples, the number of measured packets has to be scaled accordingly. For example, let $P_{A,B}$ be the portion of total traffic from an measurement point A to B. Then the number of samples at link A should be scaled up by $P_{A,B}$ to produce the required number of matched packets on average. Similarly, the number of samples at link B should be scaled up by a factor of $P_{B,A}$. Thus, the number of samples for a pair of delay measurements with the passive sampling (n^{op}_{pass}) is

$$n^{op}_{pass} = \frac{n^*}{P_{A,B}} + \frac{n^*}{P_{B,A}} \tag{10}$$

Now, we consider a network-wide number of samples in an ISP. With the passive measurement, the number of samples increases linearly with the number of measurement points, say N_{mp}, i.e., $\frac{2}{P} \cdot n^* \cdot N_{mp}$. Meanwhile, traffic fanout factor becomes inversely proportional to the number of measurement points. Let us denote the network-wide number of samples with a passive sampling as n^{nw}_{pass}. Then, for a network with N_{mp} number of measurement points and the average fanout factor P_{avg}, n^{nw}_{pass} is computed as below:

$$n^{nw}_{pass} = \sum_{(i,j)\in\{pairs\}} \left(\frac{n^*}{P_{i,j}} + \frac{n^*}{P_{j,i}} \right) \tag{11}$$

$$\approx \frac{n^*}{P_{avg}} \cdot N_{mp} \approx n^* N^2_{mp} \tag{12}$$

where we approximated the average fanout factor with the inverse of total number of measurement points in the network (i.e., $P_{avg} \approx 1/N_{mp}$).

In the active measurement, the number of samples grows linearly with the number of pair of measurement points, N_{pairs}, or quadratically with the number of measurement points. We denote the network-wide number of samples with the active measurement as n^{nw}_{actv}, and it is computed as below:

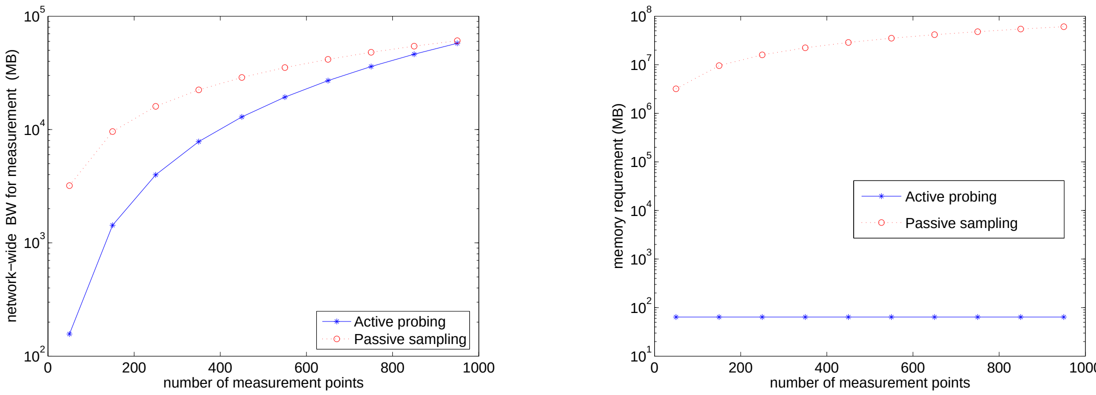
$$n^{nw}_{actv} = n^* \cdot N_{pairs} = n^* \cdot N_{mp} \cdot (N_{mp} - 1) \tag{13}$$

$$\approx n^* \cdot N^2_{mp} \tag{14}$$

To assess the amount of actual bandwidth consumed, let us assume 64 bytes are used for both an active test packet and a passive packet sample. For a passive packet sample, suppose 4 bytes for a packet hash value, 8 bytes for a timestamp, 4 bytes for a source router address, 4 bytes for a link identifier, and 20 bytes for a export protocol header (in order for a central server to recognize the measurement data) are used. Including 20 and 8 bytes for IP and UDP headers of the exported packet, it leads to a total of 64 bytes for one packet data. For a measurement interval, we also assume 1000 samples are used. Figure 13(a) illustrates the bandwidth usage for the two schemes with varying number of measurement points. The advantage of the active method is prominent when a small number of measurement points are measured. If most of the measurement points are measured in a network, the number of samples from both methods becomes similar. In practice, traffic fanout exhibits a large disparity among measurement points [8] . In addition, the fanout factor is not known in advance and varies over time, making it hard to ensure the number of samples in passive sampling. Furthermore, there may be very little or no match between the packets sampled at two measurement points. In that case, it may not be possible for a passive measurement to produce enough number of samples to obtain a delay estimate with any reasonable accuracy. On the other hand, the active method injects only a fixed, required number of samples, regardless of the traffic load between measurement points, ensuring the accuracy of the measurement. Therefore, when a portion of measurement pairs are measured, the passive sampling consumes more bandwidth , thus rendering itself more intrusive than the active measurement. In addition, passive measurement requires all packets to be hashed, potentially affecting the performance of the forwarding path of the measurement point.

Now we consider memory requirement either at a router or at a central station. In a passive measurement, sets of transferred packets have to be kept at a

[8] An instance of a PoP level traffic matrix showed that the fanout varies from 0.001% to 40% in the network of our study. The fanout factor would dramatically decrease in router or link level [21].



(a) Bandwidth consumption with measurement data

(b) Memory requirement at a measurement point or a central station

Fig. 13. Active probing vs. passive sampling: bandwidth and memory usage comparison

central station for a long enough duration of packet delay within a network. Then a set of stored packets from a measurement point will be matched with ones from another measurement point for delay computation. On the other hand, in the active measurement, each measurement point computes the delay of a test packet on arrival of the test packet at destination. Thus, only the data relevant to delay statistics (e.g., histogram) needs to be kept at a sink router. Figure 13(b) compares memory requirement of the two schemes. A fixed amount of memory is needed at a measurement point in the active scheme. In a passive measurement, however memory requirement at a central station larger than active scheme and grows with the network size.

Taking bandwidth consumption and memory requirement into consideration, for the purpose of network wide delay monitoring, we find that an active probing is more practical and less intrusive over a passive sampling.

7. Related Work

IPPM (IP Performance Metrics) [15] has defined a set of metrics [10] for measuring the quality, performance, and reliability of Internet *paths*, and developed standard *frameworks* [35] for active probing. IPPM does not provide a complete delay measurement methodology as we do. Projects such as RIPE (Reseaux IP European) TTM (Test Traffic Measurement) [29] and Surveyor [19] implement IPPM metrics, and provide GPS enabled measurement infrastructures to be deployed on networks to monitor. In these frameworks, test packet frequency is left to a user's decision.

`ping` (and its variations), `traceroute`, `pathchar` [16], `clink` [5]) are active probing tools that have not been originally designed to give accurate measures of network delay. Most of these performance measurement tools use path-oriented active probing techniques. The number of test packets and the measurement durations are typically left to user's choice. Then, average, minimum, and maximum delays are computed for the given number of test packets.

Many performance monitoring projects such as AMP (Active Measurement Project) [7], CAIDA's skitter [8], and PingER [23] employ such tools. These projects use either bursty for a short time or Poisson modulated probing. Probing frequency varies from two packets per second to one packet per hour between two measurement points. SAA [30] is an active probing tool in Cisco routers that can measure delay statistics of a path between two routers. Since the probing scheme in SAA is periodic, the statistical validity is neither known nor controllable.

Note that none of the tools or projects above has proposed an explicit delay metric and validated a test packet generation technique on real data.

A number of papers have addressed delay performance measurement. Some of them are worth mentioning, but they are not directly related to our work. End-to-end Internet delay characteristics have been studied in [2] and [27] using active test packets and/or TCP connection traces. A high precision timing technique without GPS was developed for one way delay measurement in [26]. The problem of monitoring link delays and faults that ensure complete coverage of the network are studied in [1]. In [32], authors compute delays for path segments from a set of end-to-end delay measurements by solving a system of linear equations.

Hash-based *passive* sampling in [6] proposes to use the same hashing function at all links in a network to sample the same set of packets at different links in order to infer statistics on the spatial relations of the network traffic. In [37], the author considers the problem of SLA validation with passive measurement. Given an average SLA delay value, they classify packets into two types, i.e., SLA compliant or not. It is assumed that passively measured data from two endpoints can be transferred at low load period or over a separate network.

Our work differs from all the above, in that we focus on the *representativeness* of *point-to-point* measurements, which give a concise and accurate summary of network performance for operational utilization. In particular, we investigate practical issues such as the impact of the measurement interval, the appropriate metric, boundable accuracy in delay estimation and measurement overheads. Furthermore, to the best of our knowledge, our work is the first attempt to compare and validate the performance of test packets with that of actual traffic in an operational network.

8. Conclusions

We proposed a practical delay measurement methodology designed to be implemented in operational backbone networks. It consists of measuring high quantiles (between 0.95 and 0.99) of delay over 10-30 minute time interval using pseudo random active probing. We justify each step and parameters of the technique and validate it on real delay measurement collected on a tier-1 backbone network. The accuracy of the delay measured can be controlled, and is guaranteed with a given error bound. Our method is scalable in that the number of active test packets is small, and the deployment and monitoring overhead is minimal for a backbone network measurement. We also evaluated the overhead of our active probing scheme and compared it to a passive sampling method showing active measurement becomes more practical for delay monitoring.

To the best of our knowledge, this is the first effort to propose a complete methodology to measure delay in operational networks, and validate the performance of the proposed monitoring scheme on operational data. As a part of next step, we are enhancing the methodology to monitor other performance parameters of interest to ISPs (i.e., jitter, loss, and availability).

Appendix A

Proof: [of Eq. (7)] To build a confidence interval for $\hat{q}_p$ around q_p, we first derive the relationship between $\hat{q}_p$ and q_p, in the context of random sampling. For ease of illustration, we assume that X is a continuous random variable with probability density function $f_X(x)$. As a further simplification of analysis, consider $\hat{F}(x)$ to be continuous as well. Then, note that

$$\hat{F}(\hat{q}_p) - \hat{F}(q_p) = p - \hat{F}(q_p) \tag{A.1}$$

Consider a random variable Z_i's defined as $Z_i = p - I_{X_i \le q_p}$, $(1 \le i \le n)$ Z_is are i.i.d. random variables with zero mean and a variance of $p(1-p)$. Therefore,

$$\begin{aligned} p - \hat{F}(q_p) &= \frac{1}{n}\sum_{i=1}^{n}\left(p - I_{X_i \le q_p}\right) = \frac{1}{n}\sum_{i=1}^{n}(p - Z_i) \\ &\sim N\left(0, \frac{p(1-p)}{n}\right) \end{aligned} \tag{A.2}$$

On the other hand, using a heuristic difference,

$$\begin{aligned} \hat{F}(\hat{q}_p) - \hat{F}(q_p) &\approx \hat{F}'(q_p)(\hat{q}_p - q_p) \approx F'(q_p)(\hat{q}_p - q_p) \\ &= f_x(q_p)(\hat{q}_p - q_p) \end{aligned} \tag{A.3}$$

Combining (A.1), (A.2) and (A.3), we obtain

$$\hat{q}_p \sim N\left(q_p, \frac{\sigma^2}{n}\right) \text{ where } \sigma = \frac{\sqrt{p(1-p)}}{f_x(q_p)} \tag{A.4}$$

■

References

[1] Y. Bejerano and Rajeev Rastogi. Robust Monitoring of Link Delays and Faults in IP Networks. In *IEEE INFOCOM'03*, San Francisco, March 2003.

[2] J.-C. Bolot. End-to-end packet delay and loss behavior in the Internet. In *Proceedings of ACM SIGCOMM*, San Francisco, August 1993.

[3] C. Boutremans, G. Iannaccone, and C. Diot. Impact of Link Failures on VoIP performance. In *Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*, Miami Beach, Florida, May 2002.

[4] B.-Y. Choi, S. Moon, Z.-L. Zhang, K. Papagiannaki, and C. Diot. Analysis of Point-to-Point Packet Delay in an Operational Network. In *IEEE INFOCOM'04*, Hong Kong, March 2004.

[5] A. Downey. Using pathchar to estimate Internet link characteristics. In *Proceedings of ACM SIGCOMM*, pages 241–250, Cambridge, MA, USA, October 1999.

[6] N. Duffield and M. Grossglauser. Trajectory sampling for direct traffic observation. In *Proceedings of ACM SIGCOMM*, 2000.

[7] NLANR (The National Laboratory for Applied Network Research). Active Measurement Project. http://moat.nlanr.net.

[8] CAIDA (The Cooperative Association for Internet Data Analysis). `skitter`. http://www.caida.org/tools/measurement/skitter.

[9] C. Fraleigh, S. Moon, B. Lyles, C. Cotton, M. Khan, D. Moll, R. Rockell, T. Seely, and C. Diot. Packet-level traffic measurements from the Sprint IP backbone. *IEEE Network*, 17(6):6–16, November-December 2003.

[10] G. Almes and S. Kalidindi and M. Zekauskas. A One-way Delay Metric for IPPM. *Internet Request For Comments 2679*, 1999.

[11] U. Hengartner, S. Moon, R. Mortier, and C. Diot. Detection and Analysis of Routing Loops in Packet Traces. In *ACM SIGCOMM Internet Measurement Workshop*, Marseille, France, November 2002.

[12] G. Huston. *ISP Survival Guide: Strategies for Running a Competitive ISP*. John Wiley & Sons, October 1998.

[13] G. Iannaccone, C-N. Chuah, R. Mortier, S. Bhattacharyya, and C. Diot. Analysis of link failures over an IP backbone. In *ACM SIGCOMM Internet Measurement Workshop*, Marseille, France, November 2002.

[14] Sprint ATL IPMon project. *http://ipmon.sprint.com.*

[15] IPPM. Internet Engineering Task Force, IP Performance Metric Charter. *http://www.ietf.org/html.charters/ippm-charter.html.*

[16] V. Jacobson. `pathchar`. http://www.caida.org/tools/utilities.

[17] R. Jain. *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling.* Wiley-Interscience, April 1991.

[18] S. Jamin, C. Jin, Y. Jin, R. Raz, Y. Shavitt, and L. Zhang. On the placement of Internet Instrumentation. In *Proceedings of INFOCOM*, Tel Aviv, Israel, March 2000.

[19] S. Kalidindi and M. Zakauskas. Surveyor: An Infrastructure for Internet Performance Measurements. In *Internet Networking (INET)*, San Jose, June 1999.

[20] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C-N. Chuah, and C. Diot. Characterization of failures in an IP backbone. In *IEEE Infocom*, Hong Kong, November 2004.

[21] A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, and C. Diot. Traffic Matrix Estimation: Existing Techniques and New Directions. In *Proceedings of ACM SIGCOMM*, Pittsburgh, August 2002.

[22] D. Moore. *The Basic Practice of Statistics, 3rd ed.* W. H. Freeman Publishers, April 2003.

[23] Department of Energy MICS. PingER. http://www-iepm.slac.stanford.edu/pinger.

[24] OWAMP. IETF IPPM draft: A One-Way Active Measurement Protocol. *http://www.ietf.org/internet-drafts/draft-ietf-ippm-owdp-07.txt.*

[25] K. Papagiannaki, S. Moon, C. Fraleigh, P. Thiran, and C. Diot. Measurement and Analysis of Single-Hop Delay on an IP Backbone Network. In *Proceedings of INFOCOM*, San Francisco, CA, April 2002.

[26] A. Pasztor and D. Veitch. Precision Based Timing Without GPS. In *Proceedings of ACM SIGMETRICS*, Marina Del Rey, June 2002.

[27] V. Paxson. *Measurement and Analysis of End-to-End Internet Dynamics.* PhD thesis, University of California, Berkeley, 1997.

[28] V. Paxson, A.K. Adams, and M. Mathis. Experiences with NIMI. In *Proceedings of Passive and Active Measurement Workshop*, Hamilton, New Zealand, April 2000.

[29] Paul Ridley and Karel Vietsch. A New Structure for the RIPE NCC: De Facto Organisational Rules (Revised). *RIPE-161, http://www.ripe.net/ripe/docs/ripe-161.html and references therein*, August 1997.

[30] SAA. Cisco Service Assurance Agent. *http://www.cisco.com.*

[31] R. Serfling. *Approximation Theorems of Mathematical Statistics.* Wiley, 1980.

[32] Y. Shavitt, X. Sun, A. Wool, and B. Yener. Computing the Unmeasured: An Algebraic Approach to Internet Mapping. In *IEEE INFOCOM'01*, Alaska, April 2001.

[33] *PSLA Management Handbook.* April 2002.

[34] D. Thaler and C. Hopps. Multipath issues in unicast and multicast next-hop selection. Internet Engineering Task Force Request for Comments: 2991, November 2000.

[35] V. Paxson and G. Almes and J. Mahdavi and M. Mathis. Framework for IP Performance Metrics. *Internet Request For Comments 2330*, 1998.

[36] Yin Zhang, Nick Duffield, Vern Paxson, and Scott Shenker. On the Constancy of Internet Path Properties. In *ACM SIGCOMM Internet Measurement Workshop*, San Francisco, California, USA, November 2001.

[37] T. Zseby. Deployment of Sampling Methods for SLA Validation with Non-Intrusive Measurments. In *Proceedings of Passive and Active Measurement Workshop*, Fort Collins, Colorado, April 2002.