Practical Delay Monitoring for ISPs

Baek-Young Choi
School of Computing and Engineering
University of Missouri
Kansas City Missouri, USA
choiby@umkc.edu

Sue Moon
Dept. Computer Science
Korea Advanced Institute of Science and Technology
Daejeon, Korea
sbmoon@cs.kaist.ac.kr

Rene Cruz
Dept. Electrical & Computer Engineering
University of California
San Diego CA, USA
cruz@ece.ucsd.edu

Zhi-Li Zhang
Dept. Computer Science &Engineering
University of Minnesota
Twin Cities MN, USA
zhzhang@cs.umn.edu

Christophe Diot
Intel Research
Cambridge, UK
christophe.diot@intel.com

ABSTRACT

Point-to-point delay is an important network performance measure as well as a key parameter in SLAs. 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.

Categories and Subject Descriptors

G.3 [**Probability and Statistics**]: Experimental Design, Statistical Computing

General Terms

Measurement, Performance

Keywords

Delay, Performance monitoring, Active probing

1. INTRODUCTION

Point-to-point delay is a powerful "network health" indicator. It captures service degradation due to congestion, link failure, and routing anomalies. Thus it has been used as a key parameter in Service Level Agreements (SLAs) between an ISP and its customers. Obtaining meaningful and accurate delay information is necessary for both ISPs and their customers. 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 [11, 3].

The common practice in operational 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 set the parameters of measurement tools (e.g., the probe interval, frequency and duration of measurement) in order to get a certain accuracy.

Inaccurate measurement defeats the purpose of performance monitoring. Operators may make wrong decision based on erroneous measurement data. In addition, injecting a significant number of probes 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.

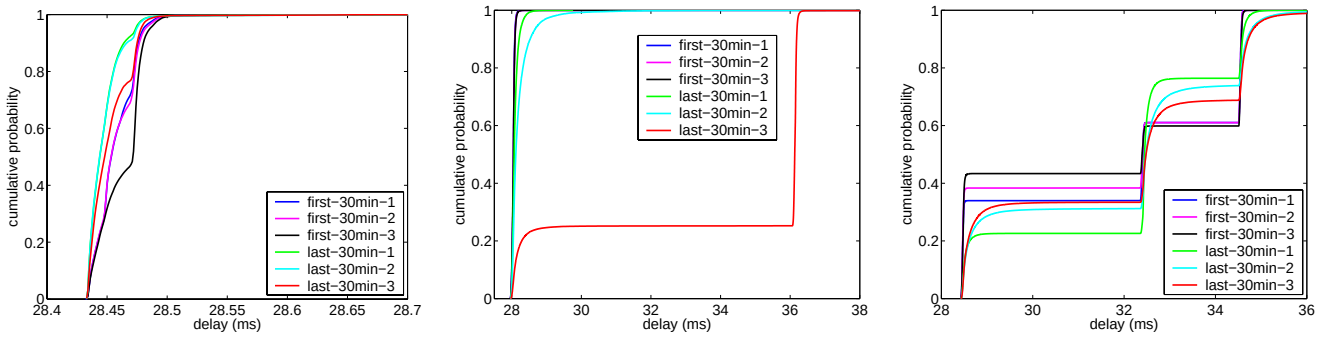
We raise the following practical concerns in monitoring delays in a backbone network. How often should delay statistics be measured? What metric(s) capture the network delay performance in a meaningful manner? How do we implement these metrics with limited impact on network performance? In essence, we 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 two-fold: (i) By analyzing the delay measurement data from an operational network (Sprint US backbone network), we identify high-

CoNEXT'05, October 24–27, 2005, Toulouse, France.

Table 1: Summary of matched traces (delay in ms)

Set	From	To	Start Time (UTC)	Duration	Packets	min.	Avg.	med.	$.99^{th}$	max.
1	OC-48	OC-12	Aug. 6. 2002 12:00	16h 24m	1,349,187	28.430	28.460	28.450	28.490	85.230
2	OC-12	OC-12	Nov. 21, 2002 14:00	5h 27m	882,768	27.945	29.610	28.065	36.200	128.530
3	OC-12	OC-48	Nov. 21, 2002 14:00	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

Figure 1: Empirical cumulative probability density function of delay over 30 minute interval

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. 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 6 we summarize related works. We conclude the paper in Section 7.

2. DATA AND BACKGROUND

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

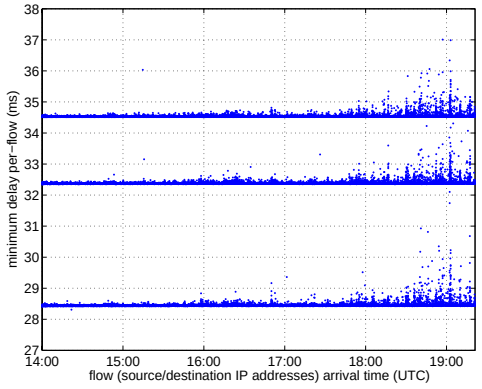


Figure 2: Presence of ECMP in Data Set 3

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 effi-

ciently 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 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 [20, 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 plot distributions for the first and last 30 minute intervals of each trace.

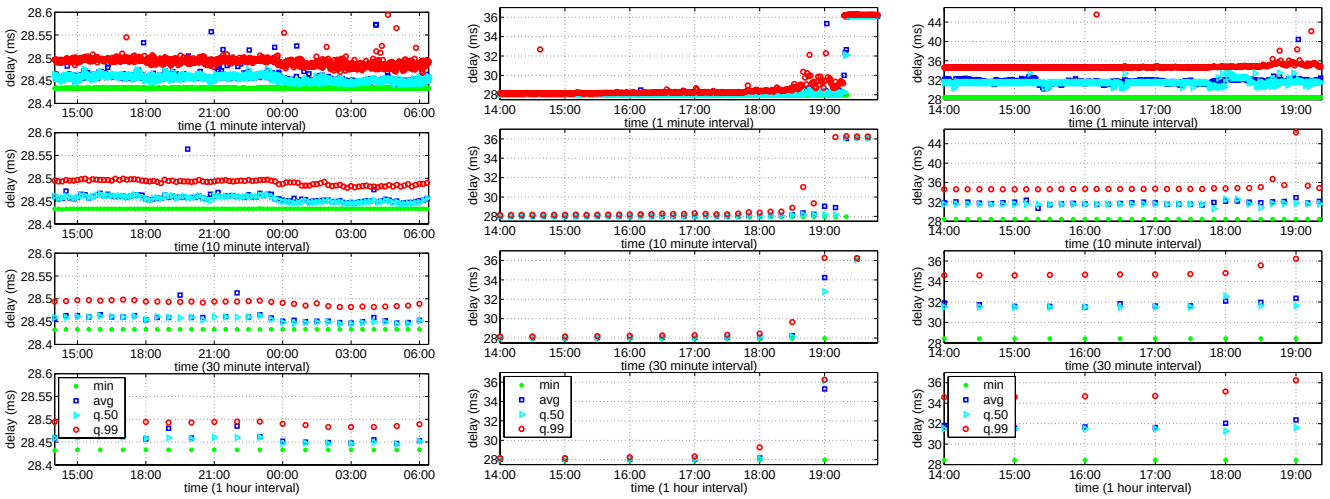
Delay distributions exhibit different shapes, as well as change over time, especially in Data Set #2 and #3. We explain these differences as follows. In theory, the packet delay consists of three components: propagation delay, transmission delay and queueing delay. Propagation delay is determined by 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 [12]. 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 [28] is commonly employed in operational networks. 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 [15], 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 intervals (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). 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



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

Figure 3: Delay metrics over different estimation intervals

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 [30] using NIMI measurement infrastructure [23] 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. The rate of outliers (IP packets with options, malformed packets, router anomalies) 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 probes 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 shortest path.
- High quantiles ($[0.95, 0.99]$) ignore the tail of the distribution and capture the 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 [16] using active probes, 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 probe 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 probes to obtain a prespecified accuracy in the estimation using Poisson modu-

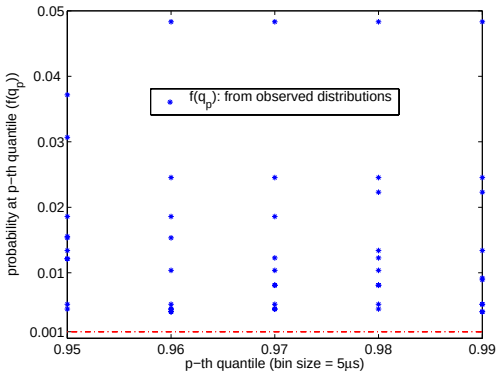


Figure 4: Empirical tail probability

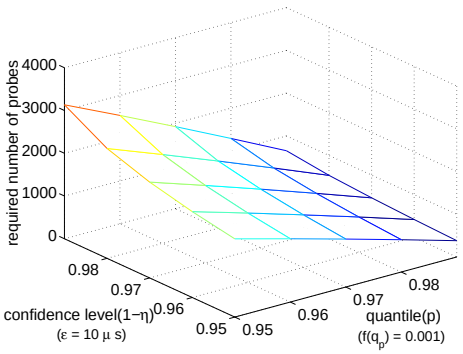


Figure 5: Number of probes required ($\varepsilon = 10\mu s$, $f(q_p) = 0.001$)

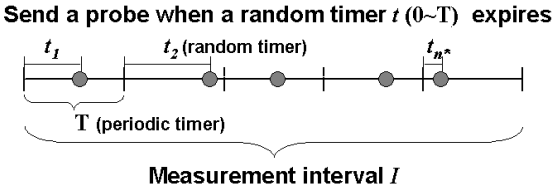


Figure 6: Scheduling n^* pseudo-random samples

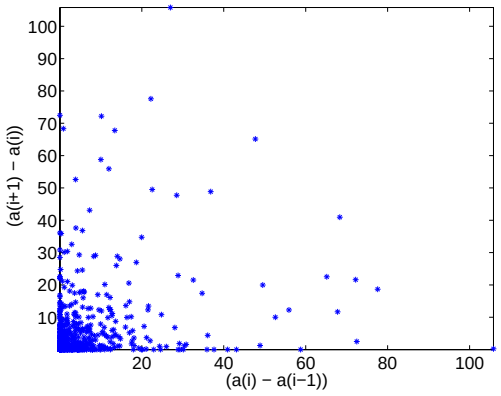


Figure 7: Correlation of inter-packet time of long delayed packets (correlation coeffcient = $1.8e-6$)

lated probing. Active probes perform like passive samples under the following two assumptions. First, the amount of probe packets should be negligible compared to the total traffic, so that it does not perturb the performance it measures. Second, the performance of probe 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 probes is relatively small, thus it is negligible on today's high speed backbone networks. Also, we encapsulate the probes 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[1]:

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

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

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

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

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

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

$$\hat{q}_p = \hat{F}^{-1}(p) \tag{3}$$

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

$$\hat{q}_p = X_{(\lceil np \rceil)} \tag{4}$$

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 $1 - \eta$:

$$Pr\{|\hat{q}_p - q_p| > \varepsilon\} \leq \eta \tag{5}$$

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

$$\hat{q}_p \xrightarrow{D} N\left(q_p, \frac{\sigma^2}{n}\right) \text{ where } \sigma = \frac{\sqrt{p(1-p)}}{f(q_p)} \quad (6)$$

$f(q_p)$ is the probability density at the p^{th} quantile of the actual distribution. Eq. (6) is called Bahadur expression [26]. 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 probes n increases, the estimate converges to the true value (i.e., $\hat{q}_p \to q_p$ as $n \to \infty$).

We derive from Eq. (5) and (6) the required number of samples to bound the estimation error within the prespecified accuracy as

$$n^* = \left\lceil z_p \cdot \frac{p(1-p)}{f^2(q_p)} \right\rceil \quad (7)$$

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

$$Var(\hat{q}_p) = \frac{p(1-p)}{f^2(q_p) \cdot n^*} \leq \frac{1}{z_p} \quad (8)$$

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

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 locations 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 4 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 probes is decided as a constant.

Figure 5 shows the number of required samples for different quantiles and different accuracy parameters. 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 probes ($420 \sim 3200$) 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 probes. This would constitute only 1.5 Kbps which is 0.0002% of the total traffic for a 30% loaded OC-48 link.

5. DELAY MONITORING METHODOLOGY

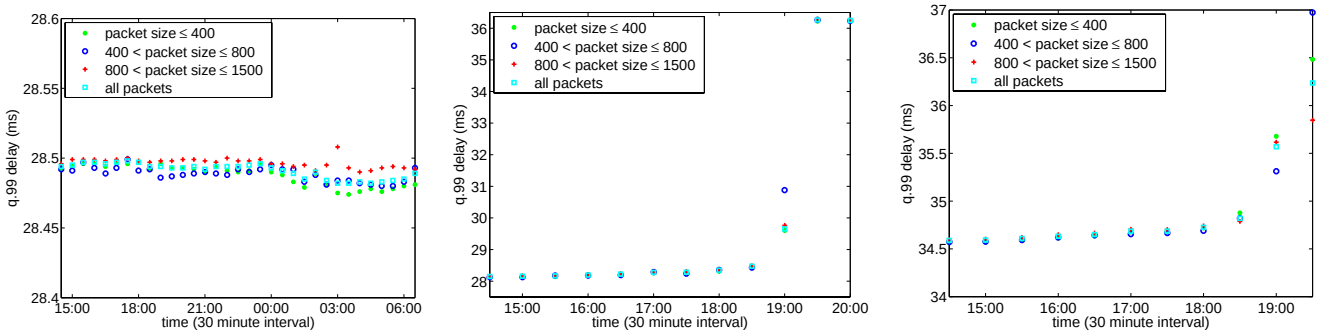
In this section, we describe our probing scheme and validate its accuracy using delay measurement data collected from the Sprint operational backbone network.

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 probes within an estimation interval I, we divide the interval into n^* subintervals of length $T(= I/n^*)$. With the help of two timers – a periodic (T) timer and a random ($t \in [0, T]$) one, a random probe is generated for each subinterval T in a time-triggered manner (i.e., whenever a random timer t expires, a probe is generated). At the end of an estimation interval (I), the delay quantile of the probe packets is computed and reported. Figure 6 illustrates graphically how to generate the pseudo-random probes. With this scheme, we ensure that n^* number of probes 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 7 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.



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

Figure 8: Impact of packet size on quantile (30 minute estimation interval)

Probe packets scheduling aside, there are several practical issues in implementing a probing scheme such as protocol type and packet size. For the type of probe 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. Probe 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 8 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 probe 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 probe 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 probe 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 probe packets to take all available paths. The original source address can be recorded in the probe payload to allow the destination to identify the source of the probes.

We have described the proposed active probing methodology in terms of probing schedule, the number of probes for a certain accuracy, the probe 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) [25][2] or IPPM one-way active measurement protocol (OWAMP) [19] can be used with little modification.

5.2 Validation

To validate the proposed technique, we emulate active probes in the following manner[3]. 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 probe measurement. The accuracy parameters are set to be $\varepsilon = 10\mu s$ [4] 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.

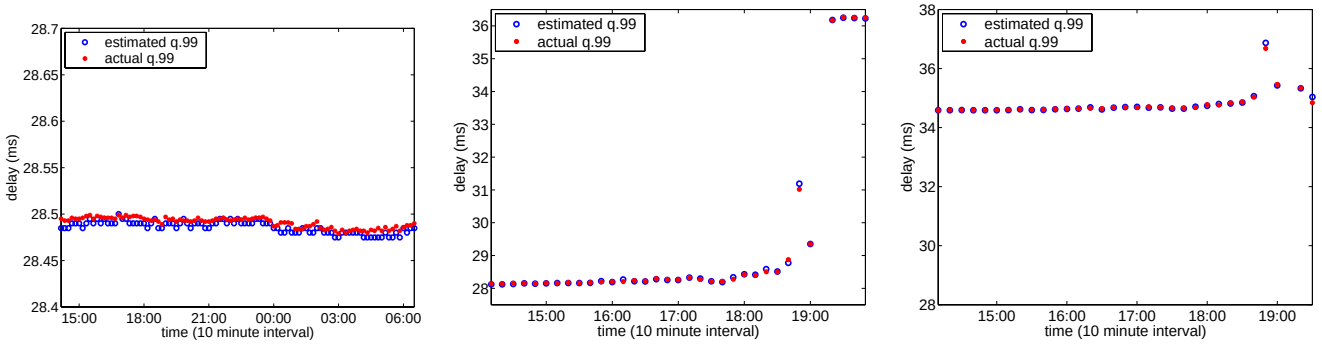
The estimated $.99^{th}$ quantiles over 10 minute intervals using 423 packets are compared with the actual $.99^{th}$ quantiles in Figure 9. Using the same number of 423 probes, the estimated quantiles are compared with the actual ones over 30 minute interval in Figure 10. Using such small numbers of 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

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

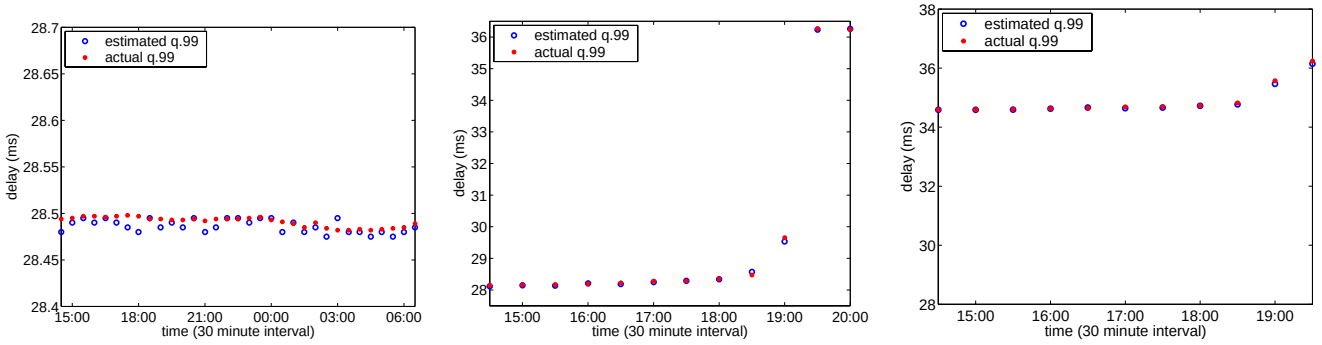
[3]We could not perform probing simultaneously to passive collection since all long-haul links on the Sprint backbone have been upgraded to OC-192 after the trace collection.

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



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

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



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

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

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

error of less than $10\mu s$, which conforms to the pre-specified accuracy parameters.

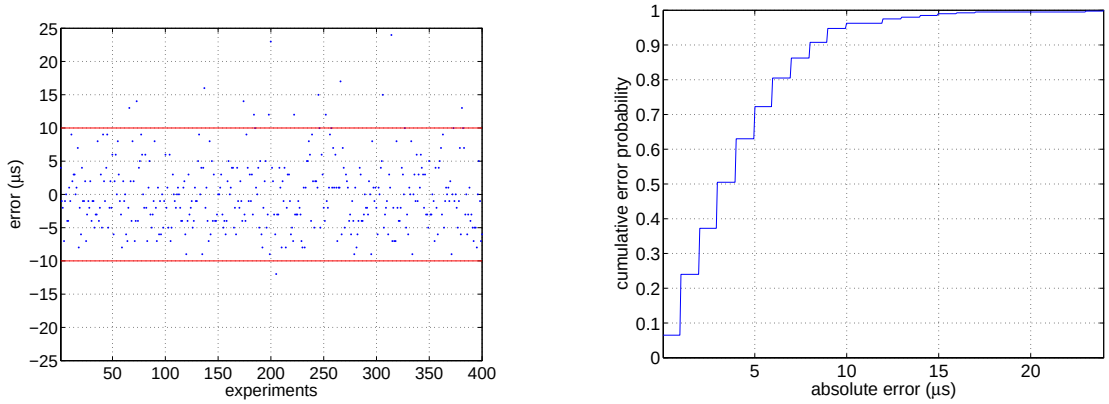
Another key metric for the performance of a sampling technique is the variance of an estimator. Small variance in estimation is a desired feature for any sampling method, as it tells the estimate is more reliable. In the previous section, we have shown that the proposed scheme enables us to bound the 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. (8) given in Section 4.

6. RELATED WORK

IPPM (IP Performance Metrics) [13] has defined a set of metrics [10] for measuring the quality, performance, and reliability of Internet *paths*, and developed standard *frameworks* [29] for active probing. IPPM does not provide a complete delay measurement methodology as we do. Projects such as RIPE (Reseaux IP Europeen) TTM (Test Traffic Measurement) [24] and Surveyor [17] implement IPPM metrics, and provide GPS enabled measurement infrastructures to be deployed on networks to monitor. In these frameworks, probe frequency is left to a user's decision.

`ping` (and its variations), `traceroute`, `pathchar` [14], `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 probes and the measurement durations are typically left to user's choice. Then, average, minimum, and maximum delays are computed for the given number of probes.

Many performance monitoring projects such as AMP (Active Measurement Project) [7], CAIDA's skitter [8], and



(a) Absolute error of estimates

(b) Cumulative error probability

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

PingER [18] 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 [25] 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 probe 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 [22] using active probes and/or TCP connection traces. A high precision timing technique without GPS was developed for one way delay measurement in [21]. The problem of monitoring link delays and faults that ensure complete coverage of the network are studied in [1]. In [27], 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 [31], 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 probes with that of actual traffic in an operational network.

7. CONCLUSIONS

We proposed a practical delay measurement methodology designed to be implemented in operational 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 probe is small, and the deployment and monitoring overhead is minimal.

To the best of our knowledge, this is the first effort to (1) propose a complete methodology to measure delay in operational networks, and (2) validate the performance of the proposed monitoring scheme on operational data.

The next step is to extend our validation by injecting active probes using our technique while measuring the real delay from passive monitoring of the link under measure. As a part of this effort, we are enhancing the methodology to monitor other performance parameters of interest to ISPs (i.e., jitter, loss, and availability).

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

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

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

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

[16] 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.

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

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

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

[20] 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.

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

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

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

[24] 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.

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

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

[27] 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.

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

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

[30] 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.

[31] 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.

APPENDIX

PROOF OF EQ. (6). 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

$$\hat{F}(\hat{q}_p) - \hat{F}(q_p) = p - \hat{F}(q_p) \tag{9}$$

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,

$$\begin{aligned} p - \hat{F}(q_p) &= \frac{1}{n}\sum_{i=1}^{n}\left(p - I_{X_i \le q_p}\right) = \frac{1}{n}\sum_{i=1}^{n}(p - Z_i) \\ &\sim N\left(0, \frac{p(1-p)}{n}\right) \end{aligned} \tag{10}$$

On the other hand, using a heuristic difference,

$$\begin{aligned} \hat{F}(\hat{q}_p) - \hat{F}(q_p) &\approx \hat{F}'(q_p)(\hat{q}_p - q_p) \approx F'(q_p)(\hat{q}_p - q_p) \\ &= f_x(q_p)(\hat{q}_p - q_p) \end{aligned} \tag{11}$$

Combining (9), (10) and (11), we obtain

$$\hat{q}_p \sim N\left(q_p, \frac{\sigma^2}{n}\right) \text{ where } \sigma = \frac{\sqrt{p(1-p)}}{f_x(q_p)} \tag{12}$$

∎