End-to-End Delay Behavior in the Internet

Li Tang[1,2], Hui Zhang[1,2], Jun Li[2,3], Yanda Li[1,3]
[1] *Department of Automation, Tsinghua University, Beijing, China*
[2] *Research Institute of Information Technology, Tsinghua University, Beijing, China*
[3] *Tsinghua National Lab for Information Science and Technology, Beijing, China*
{tangli03@mails., zhanghui04@mails., junl@, daulyd@}tsinghua.edu.cn

Abstract

While delay-critical applications typified by online multimedia communication are growing rapidly, the end-to-end (E2E) delay behavior in the Internet remains poorly understood. This paper proposes a stochastic process model, in which E2E delay alternations are classified into two categories, jump and perturbation, according to whether the statistical characterizations alter or not. As the chief type of majority delay alternations, perturbations generally occur continuously in a period, and can be analogous to an ergodic stationary process. The authenticity of the model is verified with real life delay measurements in the Internet collected by all pairs pings (APP) project through months from PlanetLab nodes. Based on the model, several delay estimation algorithms are comparatively analyzed, and the experimental results demonstrate that in terms of minimizing the mean squared error, the most accurate delay prediction is the minimum of the two most recent measurements.

1. Introduction

Understanding the E2E delay behavior has been becoming increasingly important for the further development of the Internet. A wide variety of emerging applications such as voice over IP (VoIP) and multiplayer online games are being deployed onto the Internet. These applications are highly interactive and delay-critical. For example, VoIP requires one-way domestic E2E delay no more than 150 milliseconds to achieve the same quality as PSTN telephony, not to mention some first-person-shoot online games such as Quake III necessitate delay generally under 100 milliseconds to play smoothly.

However, the Internet provides little control over the E2E delay because of its fundamental characteristic of best-effort packet switching. In the Internet, each packet generated by a source is routed through a sequence of intermediate nodes to the destination. For a specific route, the E2E delay consists of the sum of delays experienced at each hop, a physical link and its terminal node, on the way. One-hop delay depends on two kinds of factors: fixed factors including the node's capacity and the link's distance and medium, and variable factors mainly including the load of the node. The variable factors together with the alternation of routes make it difficult, if not impossible, to precisely measure the E2E delay.

The objective of this paper is to better understand the E2E delay behavior in the Internet. By analyzing the periodical round trip time (RTT) data collected by APP project [1] through months between hundreds of pairs of nodes on PlanetLab [2], we address and verify a stochastic process model to characterize the inherence of the Internet's E2E delay behavior. Based on the model, various delay estimation algorithms are comparatively studied. The results contribute to the design of mechanisms for improving the performance of the newly risen delay-critical applications.

The rest of this paper is organized as follows. Section 2 describes the data set and methodology. Section 3 presents the stochastic process model, and its verification and explication. Section 4 comparatively analyzes delay estimation algorithms. Section 5 presents related work, and Section 6 makes conclusion.

2. Methodology

2.1. Data set

Unlike bandwidth, E2E delay usually needs to be measured actively by means of probing methods rather than passively. As over-frequently probing packets may not only burden traffic but also trigger security alerts, and the time synchronization is difficult in distributed systems, the data of E2E delay in the Internet are usually collected between controlled nodes in terms of RTT. The data used in this paper are

periodical RTT measurements between PlanetLab nodes, collected by APP project from June 1 to August 31 2005. As long-term continuous delay measurements betweens nodes scattered all over the world, the data typically characterize the E2E delay behavior in the Internet.

APP project measures RTT by means of the well-known program 'ping'. Every 15 minutes, APP project makes a source node keep pinging a destination node (two adjacent ICMP echo-request packets are interspaced by 200 milliseconds), until the source node gains 10 RTT measurements, or until it times out by 120 seconds. Then, the source node records a RTT tuple consisting of the minimum, mean and maximum of the valid RTT measurements. If no RTT has ever been measured successfully before timeout, the source node will record it as a failure at that time.

2.2. Data formalization

Theoretically, the E2E delay between a pair of nodes in the Internet is a continuous process, and the data used to analyze the process should be discrete samplings of the process with a specified interval. However, each of the samples provided by the APP project is not a single value but a statistical tuple of no more than 10 consecutive RTT measurements, which raises the following questions. Can we still use the mean of the RTT measurements in each interval as the RTT sample at that time? And will this approach keeps the inherent characteristic of the E2E delay behavior? The next paragraphs clear up these concern.

Given the minimum, mean and maximum of a batch of RTT measurements in an interval, the upper bound of their standard deviation can be figured out by Theorem 1. When the divergence of a specific batch of RTT measurements is measured by the ratio of standard deviation to mean, the ratios of over 80% RTT tuples are less than 0.1. Thus, it is convincing to use the mean RTT measurements in an interval to approximately characterize the delay at that time.

Theorem 1: If the minimum, mean and maximum of arbitrary positive real numbers $x_1, x_2, \cdots, x_n$ are respectively $\underline{x}$, x, and $\overline{x}$, then the upper bound of their standard deviation is

$$u = \sqrt{\frac{1}{n-1} \cdot \left[p(\underline{x}-x)^2 + q(\overline{x}-x)^2 + (n-p-q)(y-x)^2 \right]} \quad ,$$

where $p = \left\lfloor n \cdot \frac{\overline{x}-x}{\overline{x}-\underline{x}} \right\rfloor$, $q = \left\lfloor n \cdot \frac{x-\underline{x}}{\overline{x}-\underline{x}} \right\rfloor$, and $y = n \cdot x - p \cdot \underline{x} - q \cdot \overline{x}$.

The key observations in the rest of this paper are based on the RTT time series of various node-pairs, which are formed by arranging the valid RTT samples between each pair of nodes in chronological order. Through this paper, the terms 'RTT' and 'E2E delay' are of the same meaning and interchangeably used. For a specific pair of nodes, $X(t)$ stands for the actual RTT at time t, and $\{X_1, X_2, \cdots, X_n\}$ stand for the RTT time series. Note $\{X_1, X_2, \cdots, X_n\}$ are not strictly laddered, because there are intervals in which failures rather than valid RTT tuples are recorded by APP project. In order to make the interference of spacing error as little as possible, we select the node-pairs of which the E2E delay is fairly measured in most intervals. All the RTT time series of the selected node-pairs at most miss 442 out of 8832 samples, which is negligible. For the sake of brevity, in the rest of this paper $\{X_1, X_2, \cdots, X_n\}$ will be considered as a RTT series laddered by one time unit, which in reality is equal to 15 minutes.

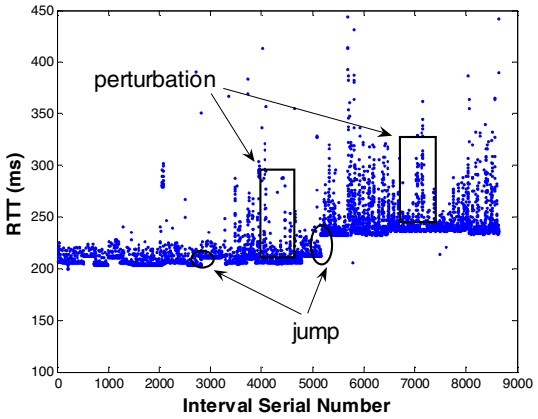

Figure 1. Time series plot of RTTs between a pair of nodes

3. Stochastic process model

3.1. Overview

As shown in Figure 1, E2E delay between a pair of nodes keep varying drastically even over short period; some sharp peaks are even twice larger than the average. Besides, there appears to be little, if any, regularity of the alternation of E2E delay, i.e. RTTs.

Generalizing from the RTT time series plots of dozens of node-pairs, we propose a model that considers the E2E delay between a pair of nodes in the Internet as a stochastic process. According to whether or not the statistical characterizations change, the model classifies the alternations of E2E delay into two categories, namely jump and perturbation. If there is no jump at all during a continuous period, the E2E delay

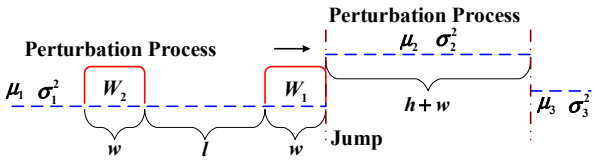

Figure 2. Sketch for RTT jump determination

in the period is named as a perturbation process, which is analogous to an ergodic stationary process.

3.2. Jump recognition

Before verifying whether or not the perturbation process can be modeled by an ergodic stationary stochastic process, it is necessary to recognize where jumps occur in a RTT time series, and cut that into a number of perturbation processes.

According to Section 3.1, a RTT jump is the breakpoint between two consecutive perturbation processes with different statistical characterizations, as shown in Figure 2. As the perturbation processes are stationary, their mathematical expectations (MEs) can be respectively denoted by μ_1 and μ_2, standard deviations by σ_1 and σ_2, and all the four parameters are constants. We make two apart windows of the same size, namely W_1 and W_2, sliding along the RTT time series, and let w denote the window size, l denote the distance between W_1 and W_2, and $h+w$ denote the duration time of the second perturbation process. The medians of the RTT samples in window W_1 and W_2, denoted by M_{W_1} and M_{W_2} respectively, are used to approximate the ME of RTT at that time. Because the distribution of RTTs is known to be single-side-heavy-tailed which will be discussed in more detail later, using the median rather than mean to approximate the ME is usually more accurate when the number of samples is small. As W_1 and W_2 slide along the RTT time series, $|M_{W_1} - M_{W_2}|$ to some extent reflects how the ME of RTTs varies with time. However, it is too assertive to judge a RTT jump whenever $|M_{W_1} - M_{W_2}|$ is larger than certain constant threshold H, because RTT perturbations can also be drastic in some periods. To better trade off misjudgment against dropout of RTT jumps, two improving approaches are utilized.

First, the threshold H is set to be $\lambda\sqrt{\sigma_{W_1} \cdot \sigma_{W_2}}$, where σ_{W_1} and σ_{W_2} respectively denote the standard deviations of the RTT samples in window W_1 and W_2, and λ is a constant parameter. In this way, the threshold accordingly rises up and comes down with the RTT perturbations becoming drastic and placid.

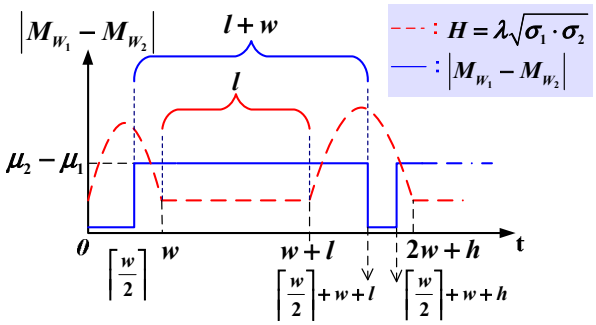

Figure 3. Time sequence diagrams when a RTT jump occurs

Second, a RTT jumps is finally judged based on the timing sequence diagrams of $|M_{W_1} - M_{W_2}|$ and H. Given that the moment demonstrating in Figure 2 is $t = 0$, Figure 3 (assuming $h > l$) presents how $|M_{W_1} - M_{W_2}|$ and H will evolve with time in future. If at sometime $|M_{W_1} - M_{W_2}|$ is larger than H, a RTT burst (more specifically, an up-burst when $M_{W_1} > M_{W_2}$, and down-burst when $M_{W_1} < M_{W_2}$) is considered to happen at that time. A RTT jump upwards/downwards around $t = \tau$ is finally judged if and only if during the period $[w, \lceil \frac{w}{2} \rceil + w + l]$, RTT up-bursts/down-bursts happen more than l times.

It is important to note that the second approach may miss one or both of two consecutive RTT jumps when the interval between them is less than $l + w$. With this problem in mind, we choose the parameters $w = 5$, $l = 3$ and $\lambda = 5$. Because RTT jumps are mainly caused by route alternation, which will be discussed in detail later, and more than 90% of routes in the Internet persist as long as several hours [3], such configuration makes the missed RTT jumps insignificant.

3.3. Stationarity of perturbation process

This subsection aims to investigate whether the perturbation process of E2E delay is stationary or not, which can be equivalently converted to examining whether or not each RTT time series of the perturbation processes is sampled from a stationary process. We use unit root tests [4] outlined in the following paragraph to achieve this purpose.

The unit root tests assume a simple autoregressive process: $y_t = \rho y_{t-1} + x_t'\delta + \varepsilon_t$, where x_t' are optional exogenous regressors which may consist of a constant,

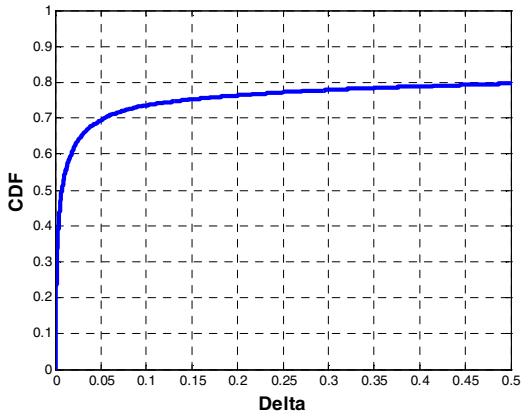

Figure 4. Cumulative distribution function (CDF) of δ value of perturbation processes

or a constant and a trend, ρ and δ are parameters to be estimated, and ε_t are assumed to be white noise. If $|\rho| \geq 1$, y is a nonstationary series and the variance of y increases with time and approaches infinity. If $|\rho| < 1$, y is a (trend-)stationary series. Thus, the hypothesis of (trend-)stationary can be evaluated by testing whether the absolute value of ρ is strictly less than one. The unit root tests generally test the null hypothesis H_0 : $|\rho| = 1$ against the one-sided alternative H_1 : $|\rho| < 1$. Depending on the data to be tested, the null hypothesis H_0 can or cannot be rejected. The result that H_0 is rejected with a significance level α can be interpreted that not the hypothesis H_0 but H_1 is true and the possibility of making a wrong decision is α.

We utilize Eviews [5], a data analysis software, to examine the stationarity of the RTT time series of perturbation processes. Specifically, we choose Augmented Dickey-Fuller (ADF) test, one of the most popular unit root tests methods, and make the exogenous regressors only include a constant, and the maximum lag be five. RTT time series of fifty node-pair's perturbation processes are tested, and the results show that all the perturbation processes can reject the hypothesis H_0 with the significance level $\alpha = 1\%$. We also tried some other unit root tests methods, such as Phillips-Perron (PP) test, and the hypothesis H_0 is also rejected with the significance level of 1% for the examined perturbation processes. Therefore, it is convincing that the perturbation process of E2E delay possesses the property of stationarity.

3.4. Ergodicity of perturbation process

The goal of this subsection is to investigate whether the perturbation process is ergodic. Strictly speaking, what to be examined is whether the time average of RTT perturbation process is equal to its ensemble average, which is also named as mean-ergodicity.

Ergodicity is an important property for a stationary stochastic process. Theoretically, to study the statistical characterizations such as the ME of a stochastic process need to repeatedly measure the variate at the same time in the same circumstances to obtain enough samples, which however is generally impossible in practice. This is because a pair of nodes can only send and receive probing packets one by one rather than processing them all simultaneously. But if a stationary process is ergodic, its statistical characterizations will be able to be studied through samples repeatedly measured at different time, because the ergodic property implies statistical homogeneity. Therefore, the ergodicity of RTT perturbation process is the basis to estimate future RTTs with historical measurements.

A necessary and sufficient condition for a wide-sense stationary process, say $\{X(t), t > 0\}$, to be ergodic is such that $\lim_{T\to\infty} \frac{1}{T}\int_0^T \left(1-\frac{\tau}{T}\right) R(\tau) d\tau = 0$, where $R(\tau)$ is the autocorrelation function and defined as $R(\tau) = \lim_{T\to\infty} \frac{1}{T}\int_0^\infty X(t)\cdot X(t+\tau)dt$. Therefore, we can investigate the ergodicity of the perturbation process by examining whether or not the RTT time series $\{X_1, X_2, \cdots, X_n\}$ can satisfy the condition $\delta = \frac{1}{n}\sum_{k=0}^{n-1}(1-\frac{k}{n})R(k) \to 0$, where $R(k) = \frac{1}{n-k}\sum_{i=1}^{n-k} X_i X_{i+k}$. RTT perturbation processes of 590 node-pairs excluding loop-back ones are investigated; Figure 4 presents the cumulative distribution function (CDF) of their δ values. Comparing to RTT samples varying from a few dozens to a few hundreds (milliseconds), most δ values are small enough to be considered as zero, for example around 70% of them are less than 0.05. Moreover, the δ values that are much greater than zero are mainly caused by the few individual fly spots (greater than 10,000), which are generally due to accidental factors. Therefore, it is convincing that the perturbation process of E2E delay indeed can be analogous to an ergodic stationary process.

3.5. Analysis and explication

To analyze the practical causes of the jump and perturbation of E2E delay, we divide the RTT into

Table 1. Main factors affecting E2E delay

	Determinative Factor	Random Factor
Terminal Delay	Node capacity Interface bandwidth	CPU load Load of multiaccess network
Switching Delay	# of passed routers * Router capacity*	Load of routers* Scheduling policies*
Propagation Delay	Link media type* Link distance*	None

three parts: 1) terminal delay caused by terminal nodes to send, receive and process the packet; 2) switching delay caused by network devices to do the same thing; 3) propagation delay caused by electric or light signal traveling through the media. The determinative and random factors affecting each parts are shown in Table 1, in which the factors depending on a specific route are marked out by asterisks.

As can be seen, given a specific route, the E2E delay mainly varies due to random factors. As the statistical characterizations of the random factors generally remains invariable with time, that is why the RTT perturbation process is stationary and ergodic. Prior work shows that most dominant route in the Internet persist a long time, which makes the perturbation to be the majority type of E2E delay alternations. On the other hand, when the route alters, several determinative factors affecting E2E delay will change, which accordingly modifies the statistical characterizations of E2E delay. In summary, the RTT perturbations are mainly due to the alternations of the load of processors and networks, while the causes for RTT jumps are mainly substantial alternations of route.

4. Estimation of E2E delay

This section studies practical algorithms estimating E2E delay according to infrequently probing packets. Recently, considerable research has proved the power of path switching and overlay routing as the means for improving E2E performance [6,7,8]. For real-time interactive communications that are sensitive to E2E delay, an accurate and portable delay estimation algorithm is the basis to make confident decisions for selecting efficient paths.

Assuming it is time $t = i$ at the moment, let a given vector $u = [X_{i-l+1}, X_{i-l+2}, \cdots, X_i]$ denote the latest l RTT samples, an unknown vector $v = [X_{i+1}, X_{i+2}, \cdots, X_{i+k}]$ denote the future k RTT samples, and a vector $\hat{v} = [\hat{X}_{i+1}, \hat{X}_{i+2}, \cdots, \hat{X}_{i+k}]$ depending on u denote the estimative values of v. Given these, the problem of delay estimation can be interpreted as to find such a mapping $f : u \to \hat{v}$ that makes the estimative error, measured by $d = < \hat{v}, v >$, a certain form of distance between $\hat{v}$ and v, as little as possible. For the sake of lucidity, we choose d as the generally used mean squared error between $\hat{v}$ and v, and define the average estimative error as $E(l,k)$, which is a function of l and k defined in Equation 1.

$$\begin{aligned} E(l,k) &= \frac{1}{n-l-k+1}\sum_{i=l}^{n-k} d < \hat{v}_i, v_i > \\ &= \frac{1}{n-l-k+1}\sum_{i=l}^{n-k}\frac{1}{k}\sum_{j=1}^{k}(\hat{X}_{i+j} - X_{i+j})^2 \end{aligned} \quad (1)$$

Therefore, a tuple of (l,k,f) uniquely stands for a RTT estimation algorithm. As the average error $E(l,k)$ of an algorithm is related to l and k, algorithms with the same l and k are evaluated by relative average error, or relative error for brevity. As shown in Equation 2, the relative error of a specific algorithm is defined as the proportion of the algorithm's error to the error caused by simply assuming all the future k RTT samples to be equal to the most recent measurement.

$$\lambda(l,k,f) = \frac{E_f(l,k)}{E_0(l,k)} = \frac{\sum_{i=l}^{n-k}\sum_{j=1}^{k}(\hat{X}_{i+j} - X_{i+j})^2}{\sum_{i=l}^{n-k}\sum_{j=1}^{k}(X_i - X_{i+j})^2} \quad (2)$$

Considering the facts that $\{X_1, X_2, \cdots, X_n\}$ consists of several RTT perturbation processes and the mathematical expectation is the best estimation of a stationary process in the sense of minimizing squared error, we respectively make use of the mean, median and minimum of the elements in u to approximate the RTT expectation during that period. Specifically, for every given l and k, three forms of mapping f_{mean}, f_{median}, and f_{min} are defined in Equation 3.

$$\begin{aligned} &f_{mean} : \hat{X}_{i+s} = \frac{1}{l}\sum_{j=i-l+1}^{i} X_j \\ &f_{median} : \hat{X}_{i+s} = \text{median}\{X_{i-l+1}, X_{i-l+2}, \cdots, X_i\} \quad (1 \le s \le k) \\ &f_{min} : \hat{X}_{i+s} = \min\{X_{i-l+1}, X_{i-l+2}, \cdots, X_i\} \end{aligned} \quad (3)$$

Table 2. Relative average error of various estimation algorithms[1]

	$k=1$			$k=2$			$k=3$			$k=4$			$k=5$		
	f_{mean}	f_{median}	f_{min}	f_{mean}	f_{median}	f_{min}	f_{mean}	f_{median}	f_{min}	f_{mean}	f_{median}	f_{min}	f_{mean}	f_{median}	f_{min}
$l=2$	.856	.856	.763	.843	.843	.730	.840	.840	.718	.839	.839	.712	.839	.839	.708
$l=3$	.832	.846	.809	.808	.800	.759	.800	.781	.737	.796	.771	.724	.794	.765	.715
$l=4$	.837	.830	.869	.804	.781	.805	.790	.760	.774	.782	.748	.755	.777	.741	.741
$l=5$	.854	.900	.931	.811	.833	.853	.792	.801	.813	.781	.782	.788	.773	.769	.769
f_{tcp}	.983			.889			.839			.807			.784		

[1] When $l=1$, the relative average error of all algorithms that are essentially the same are equivalent to 1.

Table 2 gives the average relative error of the RTT estimation algorithms with f respectively being f_{mean}, f_{median} and f_{min}, while l and k severally varying from one to five, given over the RTT time series between the 590 node-pairs. The last row of Table 2 presents the relative error of another form of mapping that is defined in Equation 4, where β is set to its commonly used value 0.9. As the iterative formula is well-known as the smoothed RTT estimation in the original TCP specification [9], we name it as f_{tcp}. It is easy to understand the relative error of f_{tcp} is independent to the value of l.

$$f_{tcp}: \hat{X}_{i+s} = Y_i \qquad (1 \le s \le k)$$
$$Y_i = \begin{cases} \beta \cdot Y_{i-1} + (1-\beta) \cdot X_i & (i > 5) \\ X_i & (i = 5) \end{cases} \qquad (4)$$

As can been seen from Table 2, for RTT estimation algorithms using f_{mean} or f_{median} as the mapping, increasing the number of historical samples, i.e. l, can make the algorithm's relative error decline at first and then bounce back. This is because the more samples are used, on the one hand the more accurately can the mean and median of the samples approximate the RTT expectation in each perturbation process, but on the other hand the greater error will be introduced when a jump occurs. The decline of the relative error at the beginning is subject to the former factor, while the subsequent rebound is subject to the latter one. It appears that the best choice for such kinds of algorithms is $l=4$ on average.

Another observation is that when l is less than or equal to four, with the same l and k, the algorithms using f_{median} get less relative error than what using f_{mean}. This is due to that the E2E delay in the Internet generally follows a Gamma-like distribution with a single-side heavy tail, in which case, the median can generally approximate the mathematical expectation more precisely than the mean, especially when the amount of samples is small. Note when l increases to five from four, the rebound of the relative error of algorithms using f_{median} is stronger than the counterpart for f_{mean}. This is comprehensible because when a jump occurs, the algorithms using f_{median} trigger greater estimative error than the ones using f_{mean}. It is correspondent with the model proposed in Section 3.

It is surprising to note that for each k, the algorithm using f_{min} with $l=2$ (predicting future RTTs equivalent to the minimum of the most recent two measurements), always obtains the least relative error out of all the evaluated algorithms in the same condition. It indicates that such an algorithm can best balance the estimative error caused by the two different categories of RTT alternations, namely jump and perturbation. According to the model presented in Section 3, two principles can help to reduce estimative error. One is to set l small in order to immediately discover a jump when it occurs. The other is to make the estimation as close as possible to the RTT's ME in a RTT perturbation process without any jump. These two principles may conflict in some sense, and the algorithm using the minimum of the most recent two measurements balances both the advantages better than other algorithms. For example, if an algorithm using f_{mean} or f_{median} sets $l=2$ according to the first principle, it will be difficult for the algorithm to precisely access the RTT's ME in a perturbation process, due to the RTT's Gamma-like distribution with a single-side heavy tail. In particular, algorithms using f_{tcp} perform worst nearly all the time.

Finally, it is important to point out that there is not a single algorithm superior to the others for all RTT time series between every node-pair. The above comparison and analysis of algorithms are actually all in the sense of statistics. As an example, Figure 5 presents the CDF plot of the relative error of four different algorithms when giving them over the RTT time series between the 590 node-pairs.

Moreover, the comparative analysis in this section generally suggests how to estimate the E2E delay i.e.

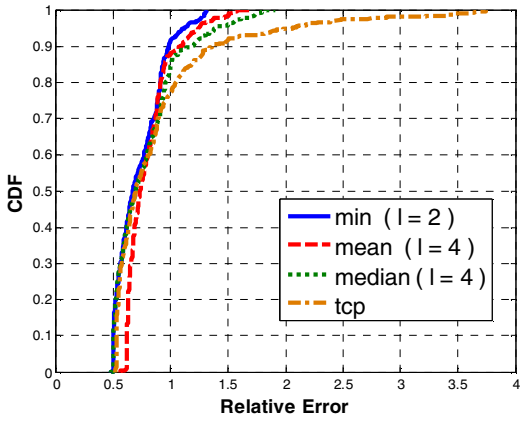

Figure 5. CDF plot of relative error of various RTT estimation algorithms given over the RTT time series of the 590 node-pairs with $k = 2$.

RTT in a short term with a few probing packets. In practical case, it is unnecessary to periodically probe every dozens of minutes; whenever the application on the source node needs to know the RTT to another node, it can probe the destination node twice with some gap, and consider the real RTT in a short term as the minimum of the two measurements. Statistically, this way is more precise than that just probing once or using the mean or median of several measurements. For the sake of long-term such as hours or longer RTT estimation, which is a seldom scenario in real life, the algorithms using f_{min} with small l are still statistically superior to other algorithms listed in Table 2; and it shows that l rises very slowly with the increase of k. For example, when k exceeds 20, the algorithm obtaining the smallest average estimative error is the one that uses f_{min} with $l = 3$, but the average estimative error of what using f_{min} with l adjacent to 3 are quite close to each other. This can be explained by the Gamma-like distribution of RTTs.

5. Related work

Systematic measurements of packet delay in the Internet can date back to the ARPANET as early as 1970s. The work included studying how delays across the ARPANET were influenced by packet length, which were used by Mills in the retransmission timeout algorithm aiming to improve TCP's performance [10]. Claffy et al. used the statistical delay data collected with an interval of 15 minutes among the NSFNET nodes to analyze the distribution of median delay [11].

Earlier 1990s, Mukherjee measured RTTs in the similar way to APP project, but Mukherjee used an interval of 1 minute rather than 15 minutes. He found the distribution of delay between most examined node-pairs could be accurately modeled by a gamma distribution plus a constant [12]. In fact, the RTT data used by Mukherjee were collected in relatively short continuous period, no more than 24 hours, in which E2E route seldom altered. Thus, what Mukherjee had pointed out was actually the distribution feature of RTTs in the same perturbation process according to the model addressed in Section 3 of this paper, and the long-term RTT distribution across several different perturbation processes should have multiple peaks. The recent work of Bovy et al [13] checked with the above argumentation. They analyzed parts of the RIPE NCC one-way delay data, and found that about 84% out of 963 normalized RTT distributions possessed typically Gamma-like shape, but there indeed existed around 5% containing multiple peaks.

Sanghi et al. ever used UDP packets with shorter interval, i.e. 39.06 milliseconds, to measure and analyze RTT. The results indicated that RTT could vary substantially over short period, and still without any deterministic regularity [14]. Bolot discussed the validity to study E2E delay by means of time series models, and proposed a simple model of a FIFO queue with finite buffer to analyze the delay and packet loss in the Internet [15].

Different from these prior works, this paper puts emphasis on setting up a model to characterize the irregular and indeterminate alternations of E2E delay in the Internet, and explore its application in delay estimation algorithms.

6. Conclusion

This paper studies the E2E delay behavior in the Internet with real life RTT data collected by APP project from globally distributed nodes on PlanetLab. A stochastic process model is proposed to reveal the characteristics of the irregular and indeterminate delay alternations in the Internet. The model classifies delay alternations into two categories, namely jump and perturbation, and indicates their natures and practical causes. The authenticity of the model is statistically verified with APP's data. Based on the model, various delay estimation algorithms are comparatively studied.

Although PlanetLab cannot represent the whole Internet, the results presented in this paper still help to better understand the Internet's E2E delay behavior, and are valuable for designing mechanisms to improve service of delay-critical applications.

Future work includes investigating the feasibility and accuracy of recognizing route alternations with E2E delay measurements.

7. Appendix: Proof of Theorem 1

What needed to prove is equivalent to the following inequation:

$$\sum_{k=1}^{n}(x_k - x)^2 \le (n-1)\cdot u^2$$
$$= p(\underline{x} - x)^2 + q(\overline{x} - x)^2 + (n-p-q)(y-x)^2$$

Let's first prove the fact that when $\sum_{i=1}^{n}(x_i - x)^2$ is maximized, there exists at most one out of $x_1, x_2, \cdots, x_n$ such that it is neither equal to $\underline{x}$ nor $\overline{x}$. (using reduction to absurdity) Assume there exist such x_i and x_j satisfying condition $x_i, x_j \notin \{x_a, x_b\}$ ($1 \le i, j \le n$ and $i \ne j$). If $x_i + x_j \ge x_a + x_b$, then let $y = x_b$ and $z = x_i + x_j - x_b$, obviously $x_a \le z \le x_b$ and $x_i^2 + x_j^2 < y^2 + z^2$; otherwise $x_i + x_j < x_a + x_b$, then let $y = x_a$ and $z = x_i + x_j - x_a$, the same there are $x_a \le z < x_b$ and $x_i^2 + x_j^2 < y^2 + z^2$. Therefore, the following inequation always holds:

$$\sum_{k=1}^{n}(x_k - x)^2 = \sum_{k=1}^{n}x_k^2 - n\cdot x^2$$
$$< \sum_{k=1,k\ne i,j}^{n} x_k^2 + y^2 + z^2 - n\cdot x^2 = \sum_{k=1}^{n}(x_k' - x)^2$$

where $x_k' = x_k$ $(1 \le k \le n, k \ne i, j)$, $x_i' = y$ and $x_j' = z$. This inequation contradicts against the prior condition that $x_1, x_2, \cdots, x_n$ already make $\sum_{i=1}^{n}(x_i - x)^2$ maximized.

Given the fact proved in the above paragraph, it is easy to obtain the inequation given in the beginning.

Another more straightforward way of proof is to convert the theorem into a constrained non-linear programming problem, and prove with Lagrange Multiplier and Kuhn-Tucker Theorem.

8. Acknowledgments

This work is sponsered by NEC Laboratories China. The authors would like to thank Dr. Yong Xia of NEC and anonymous reviewers for their helpful comments, and are grateful to Jeremy Stribling at MIT for sharing the data.

9. References

[1] All-Pairs-Pings: http://pdos.csail.mit.edu/~strib/pl_app/.

[2] PlanetLab: http://www.planet-lab.org

[3] V. Paxson, "End-to-End Routing Behavior in the Internet," *IEEE/ACM Transactions on Networking*, vol. 5, pp. 601~615, 1997.

[4] J. D. Hamilton, *Time Series Analysis*, Princeton University Press, 1994.

[5] Eviews: http://www.eviews.com.

[6] A. Akella, J. Pang, B. Maggs, S. Seshan, and A. Shaikh, "A Comparison of Overlay Routing and Multihoming Route Control," in *Proceedings of ACM SIGCOMM 2004 Conference*, 2004.

[7] S. Tao, K. Xu, Y. Xu, T. Fei, L. Gao, R. Guerin, J. Kurose, D. Towsley, and Z.-L. Zhang, "Exploring the Performance Benefits of End-to-End Path Switching," in *Proceedings of IEEE ICNP 2004 Conference*, 2004.

[8] S. Tao, K. Xu, A. Estepa, T. Fei, L. Gao, R. Guerin, J. Kurose, D. Towsley, and Z.-L. Zhang, "Improving VoIP Quality through Path Switching," in *Proceedings of IEEE INFOCOM 2005 Conference*, 2005.

[9] V. Jacobson, "Congestion Avoidance and Control," *Computer Communication Review*, vol. 18, pp. 314~329, 1988.

[10] D. L. Mills, "Internet Delay Experiments," RFC 889, 1983.

[11] K. Claffy, H.-W. Braun, and G. Polyzos, "Traffic Characteristics of the T1 NSFNET Backbone," in *Proceedings of IEEE INFOCOM 1993 Conference*, 1993.

[12] A. Mukherjee, "On the Dynamics and Significance of Low Frequency Components of Internet Load," *Internetworking: Research and Experience*, vol. 5, pp. 163~205, 1994.

[13] C. J. Bovy, H. T. Mertodimedjo, G. Hooghiemstra, H. Uiterwaal, and P. V. Mieghem, "Analysis of End-to-End Delay Measurements in the Internet," in *Proceedings of the Passive and Active Measurement Workshop-PAM*, 2002.

[14] D. Sanghi, A. Agrawala, O. Gudmundsson, and B. Jain, "Experimental Assessment of End-to-End Behavior on Internet," in *Proceedings of IEEE INFOCOM 1993 Conference*, 1993.

[15] J. C. Bolot, "Characterizing End-to-End Packet Delay and Loss in the Internet," *Journal of High-Speed Networks*, vol. 2, pp. 305~323, 1993.