Estimation of Available Bandwidth of a Remote Link or Path Segments

Seung Yeob Nam, Sihyung Lee, Hyong S. Kim

July 2, 2006
CMU-CyLab-06-012

CyLab
Carnegie Mellon University
Pittsburgh, PA 15213

Estimation of Available Bandwidth of a Remote Link or Path Segments

Seung Yeob Nam, Sihyung Lee, and Hyong S. Kim
Dept. ECE and CyLab, Carnegie Mellon University
Pittsburgh, PA 15213
synam@andrew.cmu.edu, sihyungl@cmu.edu, kim@ece.cmu.edu

ABSTRACT
Available bandwidth is usually sensitive to network anomalies such as physical link failure, congestion, and DDoS attack. Thus, real-time available bandwidth information can be used to detect network anomalies. Many schemes have been proposed to estimate the end-to-end available bandwidth or end-to-end capacity. However, available bandwidth estimation for a specific remote link has not been addressed in detail yet. We propose a new scheme to estimate the available bandwidth of remote path segments without deploying our tool at the remote nodes. We send two streams of ICMP timestamp packets to the both end nodes of the target path segment according to a Poisson process and estimate the available bandwidth for that path segment based on the measured packet delay and PASTA theory. Since 80% of routers respond to ICMP timestamp packets according to our measurement results, our scheme can monitor a much broader range of network path segments than conventional available bandwidth estimation schemes which usually require deployment of probing tools at remote nodes. We evaluate the performance of our scheme by simulation.

Keywords
Available Bandwidth Estimation, ICMP timestamp packets, Poisson Probing, PASTA, Remote Estimation, Clock Skew Correction

1. INTRODUCTION
The capacity of core networks has increased tremendously due to recent technology development in optical transmission and high-speed router/ethernet switches. However, the quality-of-service (QoS) still remains illusive for real-time multimedia services in the Internet. The congestion or network failure caused by either a real physical problem or malicious attacks such as DDoS attack [1] and worms [2] further deteriorates the QoS in the Internet. Available bandwidth is usually sensitive to network anomalies caused by link/node failure or congestion. Thus, real-time available bandwidth information can be used to detect those kinds of network anomalies.

There are several works on end-to-end available bandwidth estimation problem [3-10]. Most of them are focused on the available bandwidth of a path and the probing tools are deployed at both ends of the path. However, the internet consists of many heterogeneous sub-networks and they are not easily accessible and visible to the network operators who do not own them. It is virtually impossible to deploy the probing tool in every router and it is not easy to monitor every possible path or link with the conventional end-to-end available bandwidth estimation schemes.

We consider the problem of estimating the available bandwidth for a remote link or a path segment which consists of several consecutive links without deploying the monitoring program at remote nodes. This problem has not yet been addressed extensively in the literature. Jin et al. [11] attempted to solve a similar problem. They developed *pipechar* to estimate the available bandwidth of each link on a given path. But, it is reported that pipechar was unresponsive to variations in cross-traffic on 100 Mbps paths [12].

Popular techniques used to estimate the end-to-end available bandwidth can be usually classified into two techniques: packet pair, which is referred to as probe gap model (PGM) in [9], and packet train methods. It is not easy to extend these techniques to estimation of the available bandwidth of remote links. Especially it is very difficult to probe links beyond the *tight link* which has the minimum unused bandwidth on a given path. The reason can be explained with an example. Fig. 1 shows a path between two Nodes *A* and *B*. Let us assume that every link has a link rate of 1 Gbps except the link between Nodes n-1 and n which has a link rate of 100 Mbps. We assume that the tight link is the link between n-1 and n and we want to estimate the available bandwidth of the link between n and n+1. Usually packet pair methods assume that the corresponding queue at Node n does not become empty between arrivals of two consecutive probe packets. However, in this

case the tight link between n-1 and n tends to increase the interval between two consecutive packets by 10 times on average than other links. Thus, the above assumption is highly likely to be invalid for the links after the tight link. Packet train methods usually need to congest the target link temporarily by sending probe packets at a sufficiently high rate sometimes up to near the link rate. But, we can easily know that the probing traffic can not induce congestion at any link after the tight link because of the lowest available bandwidth at the target link.

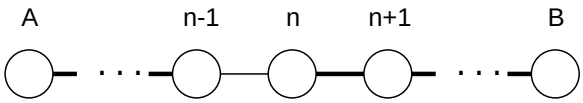

Fig. 1 Sample network path

Thus, it is not easy to estimate the available bandwidth of arbitrary remote links with conventional probing techniques. Therefore, we propose a new technique which can overcome the limitation of conventional approaches. We first estimate the delay distribution for a path segment from the monitoring node to the other remote node, and then from the obtained delay distribution we estimate the product of the available bandwidth ratio of every component link. From the ratio of the two available bandwidth ratio products, we estimate the product of the available bandwidth ratios of the links belonging to the target path segment. Since our scheme does not require any condition on the ratio of link rates of consecutive links, our scheme can estimate the available bandwidth ratio of the links beyond the tight link on a given path.

In order to measure the delay up to a remote node without deploying our program at remote nodes, we use ICMP timestamp messages. Since about 80% of routers respond to ICMP timestamp messages according to our measurement results, we expect that our scheme can be used to monitor a wide range of links around the monitoring node. However, there are two major challenges in utilizing ICMP timestamps.

First, ICMP timestamps have a rather coarse resolution of milliseconds. If we are interested only in the average delay or delay variation, then the resolution of milliseconds may not be a big problem. However, since our available bandwidth estimation scheme is dependent on the queueing delay distribution especially for small delays, coarse resolution of milliseconds is a non-negligible obstacle to our scheme. Second, the clocks on different machines are usually not synchronized and the offset between two different clocks usually changes over time. This is called *clock skew*. We find that the clock skew affects the queueing delay distribution and finally the available bandwidth ratio significantly. Although there has been previous research on the clock skew problem [13-16], we develop our own clock skew rate detection scheme in order to provide high accuracy and reliability. We address coarse resolution of ICMP timestamps and clock skew problem one after the other.

The rest of this paper is organized as follows. In Section 2, we explain how to estimate the product of the available bandwidth ratios of element links for a given path segment based on packet delays under the assumption that it is possible to measure the exact delay on the path segment for each packet. In Section 3, we consider the available bandwidth ratio estimation problem under the coarse resolution of ICMP timestamps when there is no clock skew problem. In Section 4, we describe our approach to resolve a clock skew problem between different nodes. In Section 5, we evaluate the proposed estimation scheme by simulation. Finally, conclusions are given in Section 6.

2. Basic Idea of Available Bandwidth Estimation

In this section, we explain how to estimate the ratio of available bandwidth to the link rate, which is referred to as the available bandwidth ratio hereafter, based on the packet queueing delays. In order to estimate packet queueing delays, we use ICMP packets. In this section, we assume that ICMP packets provide accurate timing information and there is no clock skew between any two different nodes.

2.1 Requirements

In order to estimate the product of the available bandwidth ratios for a target path segment or the available bandwidth ratio for a target link through ICMP timestamp packets, the following two conditions should be satisfied:

- Two end nodes of the target path segment should respond to ICMP timestamp packets.

- The routes from the monitoring node to both end nodes should overlap.

We assume that the route from the monitoring node to both the end nodes does not change during one probing period [17]. Since the duration of one probing period is usually kept not longer than 1 minute, we consider this assumption is reasonable. In a more dynamic case, our scheme can be used in conjunction with a route check scheme, e.g. traceroute.

2.2 Single Hop Case

We assume that each router in the internet can be modeled as an output-queued switch. Although commercial high-speed routers have a rather complex switching fabric with both input and output queues, their performance approaches that of output queued switches. For example, switches with the property of output queue emulation serves arriving packets in the exactly same order as the output queued switches [18, 19]. Thus, the assumption of output-queued switches is reasonable.

We consider a G/G/1 queue with an infinite size of buffer as a simplified model for a node. We assume that the queueing system is stable. Two kinds of traffic streams are offered to the system: network data traffic and *test* traffic. The test traffic is applied to monitor the status of the queueing system. Let (λ_1, μ_1) and (λ_2, μ_2) be the average arrival rate and the service rate of test traffic and network data traffic, respectively. Then, the offered load to the queue can be expressed as $\rho = \lambda_1 / \mu_1 + \lambda_2 / \mu_2$. If N denotes the number of packets in the queueing system at an arbitrary time, then we have

$$\Pr(N = 0) = 1 - \rho. \tag{1}$$

Suppose that the test traffic is offered to the queueing system according to a Poisson process. Let N^- be the number of packets in the system observed by an arriving test packet. Using PASTA (Poisson-Arrival See Time Average) [20], we have

$$1 - \rho = \Pr(N = 0) = \Pr(N^- = 0). \tag{2}$$

Let Q denote the queueing delay that a probe packet experiences in the queueing system. Then, $Q = 0$ if and only if $N^- = 0$. Thus, from (2), we can obtain

$$\Pr(Q = 0) = 1 - \rho.$$

Let *N(k)* be the number of test packets that experience zero queueing delay among k arriving test packets. Then, $\Pr(Q = 0)$ can be estimated by *N(k)/k*, and the following relation is obtained:

$$\lim_{k \to \infty} N(k)/k = 1 - \rho, \qquad a.s. \tag{3}$$

If the service rate of the system is C, then the available bandwidth of the queueing system is $C(1-\rho)$. ρ includes the offered load of test traffic (λ_1 / μ_1). But, we are interested in how much portion of the service rate is unused and available while serving the current data traffic, i.e. $C(1 - \lambda_2 / \mu_2) = C(1 - \rho + \lambda_1 / \mu_1)$. If we can keep the load of test traffic (λ_1 / μ_1) much lower than $(1-\rho)$, then we have

$$C(1 - \lambda_2 / \mu_2) \approx C(1 - \rho).$$

Thus, under the assumption that the load of test traffic is very low, we can estimate the ratio of available bandwidth to the link rate $(1-\rho)$ by counting the number of test packets which experience zero queueing delay (*N(k)*) and applying (3).

The packet delay at one hop can be decomposed into four components: processing delay, queueing delay, transmission delay, and propagation delay. If we fix the size of test packets to L, then the transmission delay is fixed to L/C. The propagation and the processing delays are assumed to be constant. The test packet will experience the minimum delay if and only if there is no other packet in the queueing system on its arrival. If an accurate arrival time (t_{in}) and an accurate departure time (t_{out}) of each test packet are provided through timestamps, then we can detect whether a test packet experiences zero queueing delay or not by comparing the difference of the two timestamp values (t_{out} - t_{in}) with the minimum delay for that hop.

2.3 Multiple Hop Case

In a real network, we may be interested not only in the available bandwidth ratio of the local link that is directly attached to the monitoring node but also in the available bandwidth ratio of the remote link that is several hops away from the monitoring node. We investigate how to estimate the product of the available bandwidth ratios of the links constituting a remote target path segment.

Let us consider the available bandwidth ratios of the links between Nodes n and $n+m$. Let us assume that Nodes n and $n+m$ are responding to ICMP timestamp messages and the path from the monitoring node (Node 0) to Node n overlaps with the path to Node $n+m$. In this case, the responsiveness of other nodes does not matter. We first discuss how to estimate the

product of the available bandwidth ratios of the links belonging to a path segment which starts from the monitoring node and finishes at the remote node n. We send a group of probe packets to Node n according to a Poisson process. Let a_0 be the packet sending time at the monitoring node. Let a_n be the value of ICMP timestamp which is assigned at the instant the corresponding probe packet arrives at Node n. We define $a_{min}(0,n)$ as

$$a_{\min}(0,n) = \min\{a_n - a_0\}.$$

$a_n - a_0$ has the value of $a_{min}(0,n)$ when there is no queueing delay at every node from *0* to *n-1*. Let N_i^- and N_i be the number of packets in the queue of Node i observed by an arriving test packet and the number of packets in the queue of Node i at an arbitrary time, respectively. If we assume that N_i^-'s are mutually independent with each other, then we have

$$\begin{aligned}
&\Pr(N_0^- = 0, N_1^- = 0, \cdots, N_{n-1}^- = 0) \\
&= \Pr(N_0^- = 0)\Pr(N_1^- = 0)\cdots\Pr(N_{n-1}^- = 0) \\
&= \Pr(N_0 = 0)\Pr(N_1 = 0)\cdots\Pr(N_{n-1} = 0) \\
&= (1-\rho_0)(1-\rho_1)\cdots(1-\rho_{n-1}),
\end{aligned} \tag{4}$$

where ρ_i is the offered load to the queue of Node i. The first equality of the above equation is obtained from the independence assumption, the second equality comes from PASTA, and the last equality comes from (1).

A random variable $Q_{0,n}$ denotes the summation of queueing delays that a probe packet experiences at transit nodes from 0 to n-1. We can easily know that $Q_{0,n} = 0$ if and only if every N_j^- is equal to zero for j = 0, 1, …, $n-1$. Thus, we have

$$\Pr(Q_{0,n} = 0) = \Pr(N_0^- = 0, N_1^- = 0, \cdots, N_{n-1}^- = 0). \tag{5}$$

Combining (4) and (5) yields

$$\Pr(Q_{0,n} = 0) = (1-\rho_0)(1-\rho_1)\cdots(1-\rho_{n-1}). \tag{6}$$

By the same reasoning, if we send probe traffic from Node 0 to $n+m$, then we can obtain the following relation:

$$\Pr(Q_{0,n+m} = 0) = (1-\rho_0)(1-\rho_1)\cdots(1-\rho_{n+m-1}). \tag{7}$$

From (6) and (7), we can use the following statistic to estimate $(1-\rho_n)\cdots(1-\rho_{n+m-1})$:

$$a(n, n+m) = \frac{\Pr(Q_{0,n+m} = 0)}{\Pr(Q_{0,n} = 0)}. \tag{8}$$

Thus, if we can measure the accurate queueing delay for each packet, we can estimate the product of the available bandwidth ratios for the target path segment in the above way using the statistic of (8).

3. Estimation of Available Bandwidth Considering Coarse Resolution of ICMP Timestamps

In the previous section, we assumed that it is possible to know the time when the test packet is sent from the monitoring node and the time when the test packet arrives at a remote node accurately. Usually for linux or unix machines, it is possible to measure packet departure time in microseconds, but the resolution of ICMP timestamps is limited to milliseconds. If a 1500 Byte packet is sent through a 1 Gbps link, then the transmission time is only 12 usec, and this implies that the order of the queueing delay can be much lower than a millisecond. Thus, the coarse resolution of timestamps is a non-trivial problem in estimating the available bandwidth ratio. In this section, we investigate how to estimate the available bandwidth ratio in the presence of a coarse resolution of ICMP timestamps. We assume that there is no clock skew problem between different nodes in this section, which is addressed in the next section.

Although the resolution of ICMP timestamps is as coarse as milliseconds, since the sending time can be measured up to microseconds, the queueing behavior can be inferred from the measured delay. We first show how the queueing delay distribution can be inferred from the measured delays of ICMP probe packets.

Fig. 2 describes some parameters related to the delay of probe packets. $a_i(p)$ is the accurate time when a probe packet p arrives at the node i. Especially when i is equal to 0, $a_0(p)$ is the time when the packet p is sent from the monitoring node 0. $D_{0,i}(p)$ is the accurate delay of the probe packet p from Node 0 to i and is defined as

$$D_{0,i}(p) = a_i(p) - a_0(p). \tag{9}$$

When $i > 0$, we can not know the accurate value of $a_i(p)$ due to the coarse resolution of ICMP timestamps. Instead, we assume that we know the accurate value of $a_0(p)$ at the sender side. $a_i'(p)$ denotes the value of the timestamp assigned to the test packet p at the instant when p arrives at Node i ($i > 0$). Then, $a_i'(p)$ is conveyed to the sender node through the *Receive timestamp* field of the ICMP timestamp reply message. If we define $D_{0,i}'(p)$ as

$$D_{0,i}'(p) = a_i'(p) - a_0(p), \tag{10}$$

then $D_{0,i}'(p)$ is a measurable metric.

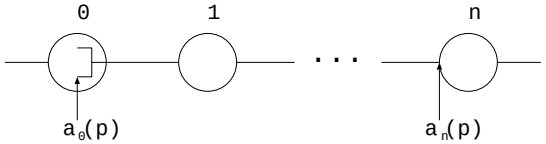

Fig. 2. Parameters related with the delay of the probe packet *p*

We next investigate the accurate delay $D_{0,i}(p)$ in more detail for available bandwidth estimation. g_i denotes the propagation delay between Nodes i and $(i+1)$ and the value of g_i is assumed to be fixed. t_i^p , q_i^p , and s_i^p denotes the transmission delay, queueing delay, and processing delay of the test packet p at Node i, respectively. Since $D_{0,i}(p)$ is the delay that the packet p experiences until it arrives at Node i, $D_{0,i}(p)$ can be expressed as

$$D_{0,i}(p) = \sum_{m=0}^{i-1} \{ s_m^p + q_m^p + t_m^p + g_m \}.$$

If we let L_p and C_i denote the size of the test packet p and the service rate of the output link of Node i, the transmission delay t_i^p can be expressed as $t_i^p = L_p / C_i$. We assume that the processing delay of packet p at Node i is fixed to s_i. When we fix the test packet size to L, we have

$$D_{0,i}(p) = D_{0,i}^f + Q_{0,i}(p), \tag{11}$$

where $D_{0,i}^f = \sum_{m=0}^{i-1} \{ s_m + g_m + L / C_m \}$, and $Q_{0,i}(p) = \sum_{m=0}^{i-1} q_m^p$.

Thus, the delay from Node 0 to i can be divided into a fixed component $D_{0,i}^f$ and a variable component $Q_{0,i}(p)$. The event that a packet p observes empty queue at every node along the path from Node 0 to i occurs if and only if $Q_{0,i}(p) = \sum_{m=0}^{i-1} q_m^p = 0$. We estimate the probability $\Pr(Q_{0,i} \le 0)$ in order to know the product of the available bandwidth ratios according to (6).

We now investigate a relation between the measurable delay $D_{0,i}'(p)$ and the queueing delay $Q_{0,i}(p)$. Fig. 3 shows the relation between $a_i(p)$ and $a_i'(p)$.

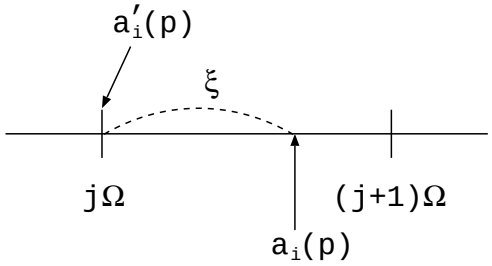

Fig. 3. Relation between $a_i(p)$ and $a_i'(p)$

$a_i'(p)$ can be expressed in terms of $a_i(p)$ as follows:

$$a_i'(p) = \Omega \lfloor a_i(p)/\Omega \rfloor,$$

where Ω is the unit of ICMP timestamp measurement, and Ω is equal to 1 msec in the current networks. If we put

$$\xi = a_i(p) - a_i'(p), \tag{12}$$

then $0 \le \xi < \Omega$. Furthermore, we can show the following regarding the distribution of ξ.

Proposition 1: If we send the probe traffic according to a Poisson process, then ξ is approximately uniformly distributed in the interval of $[0, \Omega)$.

Proof: We consider the case that we send the probe traffic according to a Poisson process. We assume that the probe packets arrive at the target node *i* according to a Poisson process. Let t_0 denote the time a minimum one-way delay later since the probing start time. In other words, t_0 is the earliest possible time the first probe packet can arrive at Node *i*. Let us assume that *k* probe packets have arrived by time *t*. Then, the arrival time of the *k* packets are uniformed distributed in the interval of $[t_0, t]$ by Theorem 2.3 in Chapter 4 of [21].

Going back to the Fig. 3, if the arrival times are uniformly distributed in the interval $[t_0, t]$ and the probing duration $t - t_0$ is a multiple of Ω, then we can easily know that ξ is uniformly distributed in the interval of $[0, \Omega)$. Even though $t - t_0$ is a not multiple of Ω, if $t - t_0$ is much larger than Ω (= 1 msec), which is usually the case, then ξ is approximately uniformly distributed in the interval of $[0, \Omega)$. ■

If we define $D_{m(0,i)}$ as the minimum value of $D_{0,i}(p)$, i.e. $D_{m(0,i)} = \min_p D_{0,i}(p)$, then from (11) we have

$$D_{m(0,i)} = D_{0,i}^f, \tag{13}$$

when $Q_{0,i}(p) = 0$. If we put $D'_{m(0,i)} = \min_p D'_{0,i}(p)$, from (10) and (12) we have $D'_{m(0,i)} = \min\{a_i(p) - a_0(p) - \xi\}$. Since $a_i(p) - a_0(p) = D_{0,i}(p)$ by (9), we have

$$\begin{aligned} D'_{m(0,i)} &= \min\{D_{0,i}(p) - \xi\} \\ &\ge \min\{D_{0,i}(p)\} - \max\{\xi\} \\ &> D_{0,i}^f - \Omega. \end{aligned}$$

Although $D_{0,i}^f - \Omega$ is a lower bound of $D'_{m(0,i)}$ by the above inequality, there may exist some packet *p* for which $D_{0,i}(p) = D_{0,i}^f$ with a non-zero probability and the difference between ξ and Ω can also decrease to 0 without bound. Thus, we approximate $D'_{m(0,i)}$ by $D_{0,i}^f - \Omega$. Since $D'_{0,i}(p) = D_{0,i}^f + Q_{0,i}(p) - \xi$ by (10), (11) and (12), we have

$$\Pr(D'_{0,i} - D'_{m(0,i)} \le x) = \Pr(Q_{0,i} - \xi + \Omega \le x), \tag{14}$$

where *x* is a non-negative real number and $D'_{0,i}$ is the random variable corresponding to $D'_{0,i}(p)$. If we put $\gamma = \Omega - \xi$, then $\gamma \sim U(0, \Omega]$ by Proposition 1 and (14) can be rewritten as

$$\Pr(D'_{0,i} - D'_{m(0,i)} \le x) = \Pr(Q_{0,i} + \gamma \le x), \tag{15}$$

(15) implies that the distribution of the measured delay $D'_{0,i} - D'_{m(0,i)}$ is equal to the convolution of the cdf of the queueing delay and a uniform pdf function of γ. Since the distribution of $D'_{0,i} - D'_{m(0,i)}$ can be obtained from the measured delay and the distribution of γ is fixed to the uniform distribution over the interval (0, Ω], we now focus on recovering the distribution of $Q_{0,i}$ based on the collected information.

We assume that $Q_{0,i}$ is independent of $\gamma (= \Omega - \xi)$. Then, (15) can be expressed as

$$\Pr(D'_{0,i} - D'_{m(0,i)} \le x) = \frac{1}{\Omega} \int_0^x \Pr(Q_{0,i} \le t) dt. \tag{16}$$

Since the probe packet sending time is usually measured in microseconds, we consider only *x* which is a multiple of Δ (Δ = 1 usec). When $x = j\Delta$, (16) can be expressed as

$$\Pr(D'_{0,i} - D'_{m(0,i)} \le j\Delta) = \frac{1}{\Omega}\sum_{i=0}^{j-1}\int_{i\Delta}^{(i+1)\Delta} \Pr(Q_{0,i} \le t)dt. \tag{17}$$

We assume that in a very short interval of $[i\Delta,(i+1)\Delta]$ $\Pr(Q_{0,i} \le t)$ can be linearly approximated as $\Pr(Q_{0,i} \le t) \approx \alpha_i t + \beta_i$. Then, we obtain

$$\int_{i\Delta}^{(i+1)\Delta} \Pr(Q_{0,i} \le t)dt \approx \frac{\Delta}{2}\{\Pr(Q_{0,i} \le i\Delta) + \Pr(Q_{0,i} \le (i+1)\Delta)\} \tag{18}$$

From (17), (18), and the piecewise linear assumption for $\Pr(Q_{0,i} \le t)$, we obtain the following relations:

$$\Pr(Q_{0,i} \le \frac{\Delta}{2}) \approx \frac{\Omega}{\Delta}\Pr(D'_{0,i} - D'_{m(0,i)} \le \Delta),$$

$$\Pr(Q_{0,i} \le (n-\frac{1}{2})\Delta) \approx \frac{\Omega}{\Delta}\{\Pr(D'_{0,i} - D'_{m(0,i)} \le n\Delta) - \Pr(D'_{0,i} - D'_{m(0,i)} \le (n-1)\Delta)\}, \qquad n \ge 2. \tag{19}$$

Using the above relations, we can estimate the distribution of $Q_{0,i}$ from the distribution of the measured delay $D'_{o,i}$.

If we know the distribution of the queueing delay $Q_{0,i}$, then we can estimate the product of the available bandwidth ratios for a path, starting from the monitoring node, by (6) or the product of the available bandwidth ratios for a path segment, starting from other node than the monitoring node, by (8). According to (6) or (8), we need to know $\Pr(Q_{0,i} \le 0)$. But, in (19) the resolution of the packet sending time Δ is not zero but 1 usec. If Δ is sufficiently smaller than the average queueing delay, then we may estimate $\Pr(Q_{0,i} \le 0)$ by $\Pr(Q_{0,i} \le \Delta/2)$ of (19).

In order to obtain a reliable value of $\Pr(Q_{0,i} \le \Delta/2)$ from the first relation of (19), a sufficient number of packets need to be sent from the monitoring node to Node *i*. The first relation of (19) can be rewritten as

$$\Pr(D'_{0,i} - D'_{m(0,i)} \le \Delta) \approx \frac{\Delta}{\Omega}\Pr(Q_{0,i} \le \frac{\Delta}{2}). \tag{20}$$

Since Δ is 1 usec and Ω is 1 msec, $\Delta/\Omega = 10^{-3}$. The left hand side of the above relation is the distribution of the delay measured under the coarse resolution (Ω) of the receiver side timestamp. Thus, (20) can be interpreted as that a minimum delay is measured when the summation of the real queueing delay $Q_{0,i}$ is zero and the probe packet arrives at Node *i* at the right time slot among Ω/Δ time slots in a 1 msec interval where the delay reduction $\xi (= D_{0,i}(p) - D'_{0,i}(p))$ due to the coarse resolution of timestamps is maximized. According to Fig. 3, the coarse resolution of receiver-side timestamps tends to make the measured delay underestimate the real delay. Thus, the measured delay is minimal when this effect is highest, i.e., when the probe packet arrives in the interval $[\,j\Omega - \Delta, j\Omega)$ for an integer *j*.

Returning to (20), even though the probability $\Pr(Q_{0,i} \le \Delta/2)$ is close to 1, $\Pr(D'_{0,i} - D'_{m(0,i)} \le \Delta) \approx \Delta/\Omega = 0.001$. In order to evaluate the probability $\Pr(D'_{0,i} - D'_{m(0,i)} \le \Delta)$, we send *k* packets from the monitoring node to Node *i*. and count the number of packets (*M(k)*) which experience the delay within Δ from the minimum delay $D'_{m(0,i)}$. We estimate $\Pr(D'_{0,i} - D'_{m(0,i)} \le \Delta)$ by *M(k)/k*. If the probability of this *minimal delay* event is 0.001, then the expected number of occurrence of that event is only once among 1000 trials. But, the probability that only one event occurs among 1000 trials is only about 0.37 under the assumption that the events are independent with each other. In this case, if the number of the minimum delay event is not 1, then the error is larger than or equal to 100% and the probability that the error is not less than 100% is 0.63. From this example, we can see that if the number of packet samples is not enough, the estimation error can be significantly large.

Formally the effect of the number of samples on the estimation accuracy can be summarized as follows. Let p_0 denote the probability that a probe packet experience the delay within Δ from the minimum delay $D'_{m(0,i)}$. We assume that each probe packet has the same minimal delay probability of p_0 and the delays of probe packets are independent of each other. Then, *M(k)* follows a binomial distribution with the parameters of (k, p_0). For a binomial random variable $X \sim$ Binomial(n, p), the following inequalities can be derived using Chernoff bounds [22]:

$$\Pr(X - np \ge \tau) \le \exp(-2\tau^2/n),$$

$$\Pr(X - np \le -\tau) \le \exp(-2\tau^2/n).$$

If we apply the above inequalities to *M(k)*, then we have

$$\Pr(|M(k)/k - p_0| \geq \varepsilon) \leq 2\exp(-2\varepsilon^2 k). \tag{21}$$

The above inequality implies that for a given ε the estimation error decreases as the number of samples k increases. Since the above bound may be loose, we do not use the above inequality to find the required number of packets k. Instead, we heuristically estimate the required number of packets so that the number of packets experiencing the minimal delay does not become too small. The detailed number is discussed later.

According to (21), the more packets we send the better estimation accuracy we can have. However, in order to collect many samples we need to send probe packets either at a high rate or during a long period. Since a high probing rate can affect the throughput of data traffic and a long probing time may prevent real-time monitoring of the available bandwidth ratio, the number of probe packets needs to be limited in real applications. Even though we send enough number of packets considering the average load, if the load on the target path segment increases significantly, *M(k)* might get too small to yield a reliable value of the available bandwidth ratio. Thus, we now consider how to estimate the available bandwidth ratio more accurately when *M(k)* is not big enough.

We focus on the available bandwidth ratio of a remote path segment, that is, a path segment which is not starting from the monitoring node. We also assume that the target path segment consists of a single link for simple explanation. $k_{0,i}$ denotes the number of probe packets sent from the monitoring node to Node i. $M_{j\Delta}(k_{0,i})$ denotes the number of packets which experience the delay within $j\Delta$ from the minimum delay $D'_{m(0,i)}$. Then, currently we estimate the available bandwidth ratio for the link between Nodes i and i+1 by

$$\hat{a}(i,i+1) = \frac{M_{\Delta}(k_{0,i+1})/k_{0,i+1}}{M_{\Delta}(k_{0,i})/k_{0,i}} \tag{22}$$

based on (8) and (20) under the assumption that $\Pr(Q_{0,i} \leq 0) \approx \Pr(Q_{0,i} \leq \Delta/2)$.

By the definition of $Q_{0,i}$ in (11), $Q_{0,i+1} = Q_{0,i} + q_i$, where q_i is the queueing delay at Node i. Since we assume that the queueing delay at Node i (q_i) is independent of queueing delays at other nodes, we have

$$\Pr(Q_{0,i+1} \leq x) = \int_0^x \Pr(Q_{0,i} \leq x - y) f_{q_i}(y) dy,$$

where $f_{q_i}(y)$ is the probability density function (pdf) of q_i. If we put $F_{Q_{0,i}}(x) = \Pr(Q_{0,i} \leq x)$, then the above equation can be expressed as

$$F_{Q_{0,i+1}}(x) = \int_0^x F_{Q_{0,i}}(x - y) f_{q_i}(y) dy, \tag{23}$$

If we model $f_{q_i}(y)$ as

$$f_{q_i}(y) = \omega_i \delta(y) + (1 - \omega_i)\tilde{f}_{q_i}(y),$$

where the function $\tilde{f}_{q_i}(y)$ is assumed to be bounded, then the available bandwidth ratio of the link between Nodes i and i+1 is $\Pr(q_i \leq 0)(= 1 - \rho_i) = \omega_i$ and (23) can be changed into

$$F_{Q_{0,i+1}}(x) = \omega_i F_{Q_{0,i}}(x) + R(x), \tag{24}$$

where $R(x) = (1 - \omega_i)\int_0^x F_{Q_{0,i}}(x - y)\tilde{f}_{q_i}(y) dy$. Since $R(x) \leq (1 - \omega_i) F_{Q_{0,i}}(x) \int_0^x \tilde{f}_{q_i}(y) dy$ and $\int_0^x \tilde{f}_{q_i}(y) dy \leq x \max_{0 \leq y \leq x} \tilde{f}_{q_i}(y)$, $\lim_{x \to 0} R(x) = 0$ and from (24) we can obtain

$$\lim_{x \to 0} F_{Q_{0,i+1}}(x) / F_{Q_{0,i}}(x) = \omega_i = \Pr(q_i \leq 0). \tag{25}$$

By (25), even though x is not zero, if x is sufficiently small, then the available bandwidth ratio of the link between Nodes i and i+1 can be estimated by

$$F_{Q_{0,i+1}}(x) / F_{Q_{0,i}}(x) = \Pr(Q_{0,i+1} \leq x) / \Pr(Q_{0,i} \leq x). \tag{26}$$

Thus, if $M_\Delta(k_{0,i})$ or $M_\Delta(k_{0,i+1})$ is too small to obtain a reliable value of $\hat{a}(i,i+1)$ in (22), then we can use the statistic

$$\tilde{a}(i,i+1) = \frac{M_{j\Delta}(k_{0,i+1})/k_{0,i+1}}{M_{j\Delta}(k_{0,i})/k_{0,i}} \tag{27}$$

to estimate the available bandwidth of the link between Nodes i and i+1 based on (26). In (26), as x increases both $\Pr(Q_{0,i+1} \le x)$ and $\Pr(Q_{0,i} \le x)$ approaches 1 and the values of both (26) and (27) also approach 1. Thus, the value of j needs to be kept as small as possible in (27).

Let us look into how small x needs to be in order to estimate the available bandwidth ratio closely with the statistic $F_{Q_{0,i+1}}(x)/F_{Q_{0,i}}(x)$ through an example. Let us consider a case where $f_{q_i}(y)$ has an exponential tail with a parameter λ, i.e.

$$f_{q_i}(y) = \omega_i \delta(y) + (1-\omega_i)\lambda e^{-\lambda y}, \tag{28}$$

From (24) and the definition of $R(x)$, we can obtain

$$F_{Q_{0,i+1}}(x)/F_{Q_{0,i}}(x) \le \omega_i + (1-\omega_i)(1-e^{-\lambda x}),$$

The term $(1-\omega_i)(1-e^{-\lambda x})$ on the right hand side of the above inequality can be considered as an upper bound of the error of the estimator $F_{Q_{0,i+1}}(x)/F_{Q_{0,i}}(x)$. Then, the range of x required to keep the error less than ζ can be obtained as

$$x < -\frac{1}{\lambda}\ln(1-\frac{\zeta}{1-\omega_i}),$$

Since the expectation of queueing delay q_i is given as $\mu_{q_i} = E[q_i] = (1-\omega_i)/\lambda$ from (28), the above inequality can be changed into

$$x < -\frac{1}{1-\omega_i}\ln(1-\frac{\zeta}{1-\omega_i})\mu_{q_i}. \tag{29}$$

Analyzing the above relation, we find that as ζ decreases the bound gets tighter. We need to note that the range of x, which corresponds to $j\Delta$ in (27), increases as the average queueing delay μ_{q_i} increases consistently as our intuition.

4. Clock Skew Rate Detection for Reliable Available Bandwidth Estimation

Thus far, we assumed that there are no clock irregularities between the monitoring node and remote nodes. But, in reality the clocks of different nodes are not always synchronized. There is usually an offset between the clocks of different nodes, and furthermore the clock offset usually changes over time, which is referred to as *clock skew*, since the clock rates are not exactly the same. Occasionally, a local clock changes abruptly due to clock reset or synchronization through either Network Time Protocol (NTP) [23] or commands in *cron* table [14]. However, the clock offset between different nodes is usually assumed to change linearly between discontinuities [13-15]. In our scheme, the probing duration is usually limited to less than 1 minute and we also assume that the clock offset between different nodes changes linearly during one probing period. In real application, the clock discontinuity can be checked with the algorithms in [13, 15]. Then, we can discard the available bandwidth estimation results for the probing periods with clock discontinuity.

In order to define clock skew in a formal way, we use the time measured by the clock on the monitoring node as the reference time t. We focus on the detection of the clock skew between the monitoring node and one destination node i. We assume that the clocks at both sender and receiver nodes progress continuously. t_j denotes the time when the j-th probe packet is sent from the monitoring node and $\tilde{a}(t_j)$ denotes the arrival time of the j-th probe packet at the destination node i according to the clock at node i. Let $z(t)$ denote the value of clock at the destination node at time t according to the sender clock. We model the clock rate difference as follows:

$$z(t) = (1+\alpha)t + \beta.$$

When $\alpha \neq 0$, the clock offset between two nodes changes over time and α is referred to as the *clock skew rate*. If the j-th probe packet sent at time t_j experiences a fixed delay of D^f and a total queueing delay of $Q(j)$ according to (11), then we have

$$\tilde{a}(t_j) = z(t_j + D^f + Q(j))$$
$$= (1+\alpha)(t_j + D^f + Q(j)) + \beta,$$

and the measured packet delay $d(t_j)$ is given by

$$d(t_j) = \tilde{a}(t_j) - t_j$$
$$= \alpha t_j + (1+\alpha)Q(j) + (1+\alpha)D^f + \beta.$$

Fig. 4 shows a sample delay sequence measured between two different nodes in the Internet. Our goal is to accurately estimate the clock skew rate α in order to remove or minimize the effect of clock skew on the queueing delay distribution.

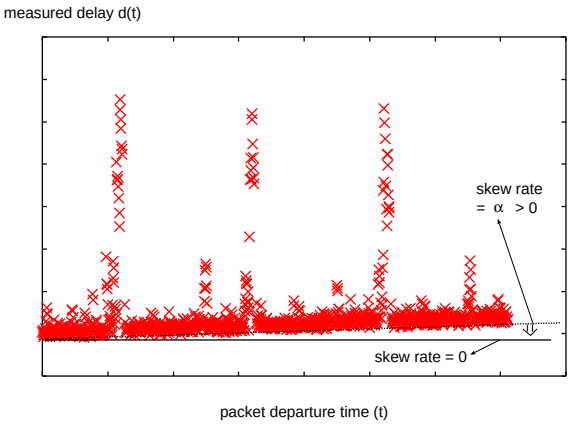

Fig. 4. Sample measured delay sequence exhibiting clock skew

Let us consider the case that we correct the clock skew correctly, that is, we correct the effect of clock skew on the measured delay as

$$\tilde{d}(t_j) = d(t_j) - \alpha t_j$$
$$= (1+\alpha)Q(j) + (1+\alpha)D^f + \beta.$$

Since $|\alpha| << 1$, we have $\tilde{d}(t_j) \approx Q(j) + D^f + \beta$. Since the minimum value of $\tilde{d}(t_j)$ is $D^f + \beta$, we can infer the queueing delay distribution from $\tilde{d}(t_j) - \min_j d(t_j)$.

If we correct the clock skew by

$$d'(t_j) = d(t_j) - \alpha' t_j$$
$$= (\alpha - \alpha')t_j + (1+\alpha)Q(j) + (1+\alpha)D^f + \beta, \quad (30)$$

then the absolute value of the first term increases linearly with respect to the packet sending time when $\alpha \neq \alpha'$. We assume that the probing is started at time 0 by the sender side clock without loss of generality. Then, let us consider a probing period of [0, *T*]. Let *K* denote the number of packets sent during this probing period. We assume that sufficiently many packets are sent such that $t_1 \approx 0$ compared with *T*.

If $\alpha' \leq \alpha$, then $\min_j d'(t_j)$ is likely to be obtained for a small value of *j* since $(\alpha - \alpha')t_j$ monotonically increases as *j* increases. If we also assume that the offered load is sufficiently low to yield $Q(j) = 0$ frequently for many *j*'s, then we may estimate $\min_j d'(t_j)$ as $(1+\alpha)D^f + \beta$ from (30). Since usually $|\alpha| << 1$, for $x \geq 0$ and $\alpha_1 \leq \alpha$ we have

$$\Pr(d'(t_l) - \min_j d'(t_j) \leq x \mid K = k, \alpha' = \alpha_1)$$
$$\approx \Pr((\alpha - \alpha_1)t_l + Q(l) \leq x \mid K = k, \alpha' = \alpha_1) \quad (31)$$
$$= \Pr((\alpha - \alpha_1)t_l + Q(l) \leq x \mid K = k),$$

where the last equality is obtained since α' is assumed to be independent of t_l and $Q(l)$. Let d' denote the delay of an arbitrary probe packet among K packets obtained after skew correction. Then, from (31) we have

$$\Pr(d' - \min_j d'(t_j) \le x \mid \alpha' = \alpha_1) = \sum_{k=1}^{\infty} \Pr(d' - \min_j d'(t_j) \le x \mid K = k, \alpha' = \alpha_1) \Pr(K = k \mid \alpha' = \alpha_1). \tag{32}$$

Since α' and K are independent of each other, $\Pr(K = k \mid \alpha' = \alpha_1) = \Pr(K = k)$, and from (31) we have

$$\begin{aligned} \Pr(d' - \min_j d'(t_j) \le x \mid K = k, \alpha' = \alpha_1) &= \sum_{l=1}^{k} \frac{1}{k} \Pr(d'(t_l) - \min_j d'(t_j) \le x \mid K = k, \alpha' = \alpha_1) \\ &\approx \sum_{l=1}^{k} \frac{1}{k} \Pr((\alpha - \alpha_1) t_l + Q(l) \le x \mid K = k). \end{aligned} \tag{33}$$

Combining (32) and (33) yields

$$\Pr(d' - \min_j d'(t_j) \le x \mid \alpha' = \alpha_1) \approx \sum_{k=1}^{\infty} \Pr(K = k) \frac{1}{k} \sum_{l=1}^{k} \Pr((\alpha - \alpha_1) t_l + Q(l) \le x \mid K = k), \tag{34}$$

where $Q(l)$ can be assumed to be independent of K. Since t_l is non-negative, for $\alpha_2 < \alpha_1 (\le \alpha)$, $(\alpha - \alpha_1) t_l \mid_{K=k} + Q(l) \le (\alpha - \alpha_2) t_l \mid_{K=k} + Q(l)$, and thus, we have

$$\Pr((\alpha - \alpha_1) t_l + Q(l) \le x \mid K = k) \ge \Pr((\alpha - \alpha_2) t_l + Q(l) \le x \mid K = k). \tag{35}$$

From (34) and (35), we can obtain

$$\Pr(d' - \min_j d'(t_j) \le x \mid \alpha' = \alpha_1) \ge \Pr(d' - \min_j d'(t_j) \le x \mid \alpha' = \alpha_2). \tag{36}$$

(36) implies that when α' is less than or equal to α in (30), as α' approaches α the measured delay values are more concentrated near the minimum value. In other words, as α' approaches α from the left, the θ-quantile delay is monotonically decreasing. When $\alpha' \ge \alpha$ in (30), we can also show that the θ-quantile delay is monotonically increasing as α' increases in a similar way. In summary, since θ-quantile delay is minimized when $\alpha' = \alpha$, using this property we can detect the clock skew rate α.

Fig. 5 briefly describes the clock skew rate (α) detection algorithm. We first set a possible range of α. Through numerous tests, we found that the absolute value of α was usually of the order of 10^{-6} and it rarely exceeded 10^{-3} except the case of abrupt changes due to clock adjustment. Thus, we set -10^{-3} as the initial lower bound of α (α'_{low}) and 10^{-3} as the initial upper bound of α (α'_{high}). We then partition the interval $[\alpha'_{low}, \alpha'_{high}]$ evenly into $2u$ subintervals as shown in Fig. 5. We then evaluate θ-quantile delay for $\alpha' = \hat{\alpha}(i)$, where $\hat{\alpha}(i) = \alpha'_{low} + i(\alpha'_{high} - \alpha'_{low})/2u$ and $i = 0, 1, \cdots, 2u$. If the minimum value of θ-quantile delay is obtained $\alpha' = \hat{\alpha}(j)$, then α should be in the interval $[\max\{\alpha'_{low}, \hat{\alpha}(j-1)\}, \min\{\alpha'_{high}, \hat{\alpha}(j+1)\}]$. If α exists outside of this interval, then there must be at least one local minimum in the interval $[\max\{\alpha_{low}, \hat{\alpha}(j-1)\}, \min\{\alpha'_{high}, \hat{\alpha}(j+1)\}]$, which contradicts the property that the θ-quantile delay is monotonically decreasing on the left side of α and monotonically increasing on the right side α. Thus, α should belong to $[\max\{\alpha'_{low}, \hat{\alpha}(j-1)\}, \min\{\alpha'_{high}, \hat{\alpha}(j+1)\}]$. Finding an interval that includes α completes the first iteration and the next iteration can be done by repeating the whole process with α'_{low} and α'_{high} modified to $\alpha'_{low} = \max\{\alpha'_{low}, \hat{\alpha}(j-1)\}$ and $\alpha'_{high} = \min\{\alpha'_{high}, \hat{\alpha}(j+1)\}$. α is estimated by $\alpha' = \hat{\alpha}(j)$ which minimize the value of θ-quantile delay in the last iteration.

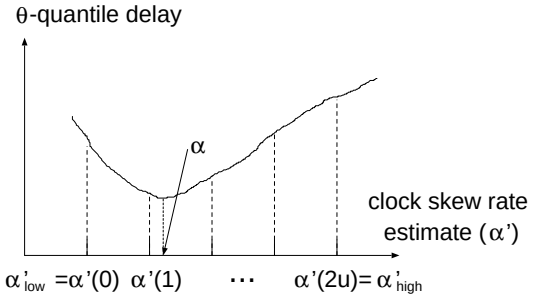

Fig. 5. Clock skew rate (α) detection algorithm

We only consider the partition of the interval $[\alpha'_{low}, \alpha'_{high}]$ into an even number of subintervals since if we calculate θ-quantile delay for $\alpha' = \hat{\alpha}(i)$, $i = 0,1,\cdots,2u$, then usually three θ-quantile delay values corresponding to $\alpha' = \hat{\alpha}(j-1), \hat{\alpha}(j)$, and $\hat{\alpha}(j+1)$ can be directly used in the next iteration due to overlapping, where $\hat{\alpha}(j)$ is the clock skew estimate which minimize the θ-quantile delay in the current iteration. The interval $[\alpha'_{low}, \alpha'_{high}]$ obtained at the end of iterations is referred to as the *estimate range* of α. We consider the worst case in terms of the number of θ-quantile delay calculations required to achieve a given length of an estimate range. In the worst case scenario, $\hat{\alpha}(0)$ and $\hat{\alpha}(2u)$ are never selected as the optimal value of α' in any iteration since their selection tends to decrease the resulting interval $[\alpha'_{low}, \alpha'_{high}]$ more (by twice) than the otherwise case. In the worst case scenario, the number of θ-quantile delay calculations (N_{del_cal}) performed up to the i-th iteration is

$$N_{del\_cal} = (2u+1) + (i-1)(2u-2) = (2u-2)i + 3. \tag{37}$$

If we let I_0 denote the length of the initial interval $[\alpha'_{low}, \alpha'_{high}]$, then after i iterations the length of $[\alpha'_{low}, \alpha'_{high}]$ is decreased to $(2/2u)^i I_0 = I_0 / u^i$. We now calculate the number of iterations required to achieve the estimate range length less than or equal to εI_0 ($\varepsilon < 1$). By solving $I_0 / u^i \le \varepsilon I_0$, we obtain

$$i \ge \frac{-\log(\varepsilon)}{\log(u)}. \tag{38}$$

Then, the number of θ-quantile delay calculations (N_{del_cal}) required to achieve the estimate range of not larger than εI_0 is obtained from (37) and (38) as

$$N_{del\_cal} \ge \frac{-\log(\varepsilon)(2u-2)}{\log(u)} + 3. \tag{39}$$

We consider only u larger than or equal to 2 and the right hand side term of the above inequality increases with respect to u larger than or equal to 2. Thus, N_{del_cal} in (39) is minimized when u = 2 and hereafter we fix the value of u to 2. If we consider 10^{-7} for the value of ε in (38), then we obtain $i \ge 24$ from (38) since u is fixed to 2. Thus, the number of iterations is fixed to 24.

If the clock skew rate α is finally estimated, then we get rid of the component due to the clock skew rate from the measured delay as shown in Fig. 4 and we apply the available bandwidth estimation scheme developed in the previous section to the skew-corrected delay sequence.

5. Numerical Results

In this section, we first investigate the effect of clock skew estimation error on the available bandwidth estimation performance and the accuracy of our clock skew rate estimation scheme. Then, we evaluate the performance of the proposed available bandwidth estimation scheme through OPNET simulation.

Fig. 6 shows the network topology used for OPNET simulation. All the link rates are fixed to 1 Gbps. Ingress router *IR1* is ten hops away from the egress router *ER1* and every intermediate core router *CRi* ($i = 1,2,\cdots,9$) and *ER1* are responding to ICMP timestamp packets. Source node *Si* sends cross traffic to Destination node *Di* through the shortest path, e.g. the cross traffic from Node *S2* to Node *D2* follows the path *S2 - CR1 - CR2 - D2*. For cross traffic, we use two types of traffic patterns, Poisson and self-similar traffic. A self-similar traffic pattern is used since the traffic patterns of today's IP networks are known to exhibit self-similarity and long-range dependence [24-26]. We use a multi-fractal model [27] to generate the self-similar traffic pattern and the hurst parameter is set to 0.8. The size of each packet of cross traffic is selected from the following distribution: 40 bytes – 60%, 576 bytes – 20%, 1500 bytes – 20%.

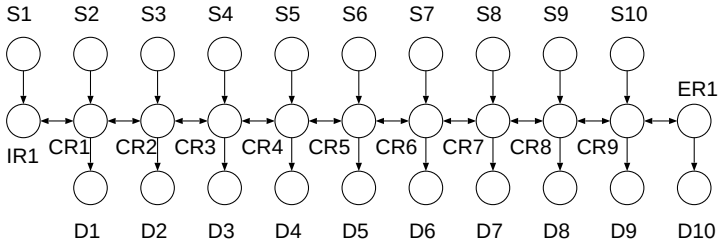

Fig. 6. Simulation network topology

We already noted that the estimator for the available bandwidth ratio (22) can be very unreliable especially when $M_{\Delta}(k_{0,i})$ or $M_{\Delta}(k_{0,i+1})$ is too small. If we send and receive the same number of packets to Nodes *i* and *i*+1, i.e. $k_{0,i} = k_{0,i+1}$, (22) is simplified to $\hat{a}(i,i+1) = M_{\Delta}(k_{0,i+1}) \ / M_{\Delta}(k_{0,i})$ and from this we can easily know why $M_{\Delta}(k_{0,i})$ and $M_{\Delta}(k_{0,i+1})$ need to be large enough. In order to cope with the case where $M_{\Delta}(k_{0,i})$ or $M_{\Delta}(k_{0,i+1})$ is too small, we suggested another estimator $\tilde{a}(i,i+1)$ in (27). When $k_{0,i} = k_{0,i+1}$, we can obtain a similar form of

$$\tilde{a}(i,i+1) = \frac{M_{j\Delta}(k_{0,i+1})}{M_{j\Delta}(k_{0,i})}. \tag{40}$$

In order to improve the reliability of the estimator by reserving enough packet counts in both numerator and denominator of the above estimator, we decide *j* in the following way:

$$j = \min\{\, j' \mid M_{j'\Delta}(k_{0,i+1}) \geq M_{th}, M_{j'\Delta}(k_{0,i}) \geq M_{th} \}, \tag{41}$$

where M_{th} is a threshold used to determine an appropriate minimum value of numerator or denominator of (40). We find that the value of M_{th} in the range of 45 to 50 usually yields good results from many simulations and we select 47 among them and use hereafter.

If we assume that $\Pr(Q_{0,i} \leq 0) \approx \Pr(Q_{0,i} \leq \Delta/2)$, then (20) can be expressed as

$$\Pr(D'_{0,i} - D'_{m(0,i)} \leq \Delta) \approx \frac{\Delta}{\Omega} \Pr(Q_{0,i} \leq 0). \tag{42}$$

Combining (6) and (42) yields

$$\Pr(D'_{0,i} - D'_{m(0,i)} \leq \Delta) \approx \frac{\Delta}{\Omega}(1-\rho_1)(1-\rho_2)\cdots(1-\rho_{i-1}).$$

Since $M_{\Delta}(k_{0,i})/k_{0,i}$ converges to $\Pr(D'_{0,i} - D'_{m(0,i)} \leq \Delta)$ as $k_{0,i}$ goes to infinity, the above relation can be rewritten as

$$\frac{M_{\Delta}(k_{0,i})}{k_{0,i}} \approx \frac{\Delta}{\Omega}(1-\rho_1)(1-\rho_2)\cdots(1-\rho_{i-1}). \tag{43}$$

If we are to send probe packets so that *j* can be 1 in (41), i.e. $M_{\Delta}(k_{0,i}) \geq M_{th}$, then the minimum number of $k_{0,i}$ can be determined from (43) as

$$k_{0,i} \geq \frac{\Omega}{\Delta} \frac{M_{th}}{(1-\rho_1)(1-\rho_2)\cdots(1-\rho_{i-1})}. \tag{44}$$

From the above inequality, we can easily know that more packets need to be sent to probe farther links. However, the probing rate needs to be limited in order to prevent the degradation of data traffic performance due to probe traffic. In addition, the probing duration also need to be limited if we want to check the target link status frequently. If r_m and v_m denote the maximum limits on the probing rate and the probing duration, respectively, then $k_{0,i}$ is limited by

$$k_{0,i} \leq \frac{r_m v_m}{L}, \tag{45}$$

where *L* is the probe packet size and *L* is fixed to 40 bytes. (44) and (45) implies when $k_{0,i}$ is limited, the scope of our probing scheme, i.e. *i* in (44), is likely to be limited. However, it is reported that link utilizations are usually less than 50% [28]. Thus, as long as link utilizations stay rather low, our scheme may be used to probe links that are moderate number of hops away from the monitoring node. We put r_m = 2 Mbps and v_m = 60 seconds in this section. Two 2 Mbps streams, one for

each end node, are occupying only 0.4% of the 1Gbps link and we assume that the effect of the probing traffic load on the performance of data traffic is negligible.[1] The probing rate and the probing duration are set to r_m and v_m if not specified otherwise. Since the probe packet size is fixed to 40 bytes, about 3.8×10^5 packets are sent during one probing period.

We now investigate the effect of clock skew rate estimation error on the available bandwidth estimation accuracy. In Fig. 6, the clock at *IR1* is the reference clock and the clock skew rate (α) of Node *CRi* ($i=1,2,\cdots,9$) is set to $(i+1)\times5.0\times10^{-7}$. The clock skew rate of *ER1* is set to 5.5×10^{-6}. We send 300 Mbps of Poisson cross traffic from *Si* to *Di* ($i=1,2,\cdots,10$). Thus, the available bandwidth ratio of the links on the path from *IR1* to *ER1* is 0.7. In this simulation scenario, the target link is the 7-th link between *CR6* and *CR7* in Fig. 6. In order to avoid a mixed effect of clock skew estimation errors on both *CR6* and *CR7*, we assume that the accurate value of the clock skew rate of *CR6*, 3.5×10^{-6}, is known in advance. The value of θ is set to 0.9, but we find that the accuracy of the proposed clock skew estimation scheme is almost independent of the detailed values of θ. Under these conditions, Fig. 7 shows the effect of clock skew estimation error for *CR7* on the available bandwidth estimation accuracy. We run 10 simulations for each case and calculate the mean μ and the standard deviation σ of the estimated available bandwidth. The *Estimated ABW* shows the range of $[\mu-\sigma,\mu+\sigma]$ of the estimated available bandwidth for each error case. We observe that the available bandwidth estimation scheme tends to underestimate the available bandwidth ratio significantly as the absolute value of clock skew rate estimation error ($|\alpha'-\alpha|$) increases. This can be explained as follows. The θ-quantile delay increases as $|\alpha'-\alpha|$ increases. The increase of θ-quantile delay implies that the pdf of the measured delay is more spread decreasing the numerator of (27). Thus, the estimator value also tends to decrease.

The two rectangular regions in Fig. 7 compare the performance of the proposed clock skew rate correction scheme with that of Moon *et al.*'s linear programming-based scheme [14].[2] The width of each rectangular region represents the range of clock skew rate estimation error obtained from 10 simulations for each scheme. The height of each rectangular region represents the range $[\mu-\sigma,\ \mu+\sigma]$ of the estimated available bandwidth obtained with each clock skew correction scheme. From the fact that the available bandwidth estimation range of the proposed clock skew detection scheme is very similar to that for the case of $\alpha'-\alpha=0$, we find that our clock skew detection scheme is rather accurate. We also observe that the proposed clock skew detection scheme has accuracy similar to the linear programming-based scheme. The proposed clock skew detection scheme has the same complexity of *O(K)* as the linear programming-based approach, where *K* is the number of the probe packets.

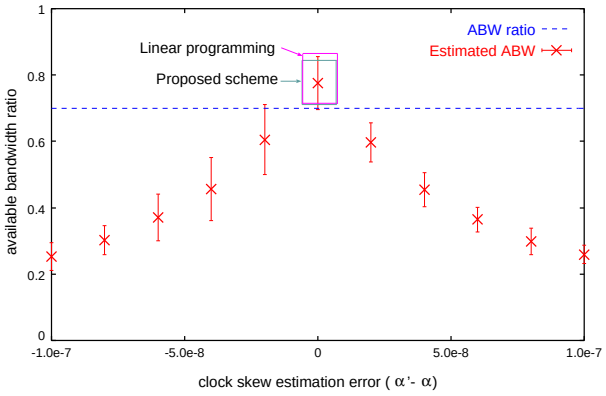

Fig. 7. Effect of clock skew rate estimation error on the accuracy of the available bandwidth estimation scheme

The difference between the proposed clock skew detection scheme and the linear programming-based scheme can be described with the following example. Fig. 8 shows a sample sequence of measured delays with respect to each packet departure time. Since there is no increasing or decreasing tendency on the measured delays, it is highly likely that there is no clock skew between the corresponding node pair. The linear programming method [14] attempts to minimize the sum of distances (along the y-axis) between the clock skew estimation line and the data points under the condition that the resulting

[1] When the link rate is lower than 1 Gbps, the probing rate can also be decreased accordingly. The reason can be explained as follows. If the link rate decreases, then the average delay increases because of the low link speed under the same offered load ρ. Then, j of $j\Delta$ can be increased by (29). Thus, the probing rate can be lowered accordingly.

[2] Convex hull algorithm in [15] is essentially the same as the linear programming algorithm in [14] when there is no clock reset.

delays do not become negative after the skew is removed. If a line passes the point (0.005, 0.001) and the slope of the line is between -0.25 and 0.25, then that line minimizes the sum of the distances noted above. Thus, an infinite number of solutions exist for this case according to the linear programming method. However, in our scheme the θ-quantile delay ($\theta \geq 4/9$) is minimized only when the clock skew estimation line has a slope of zero.[3] Thus, our scheme estimates the clock skew rate more accurately than the linear programming-based scheme in this case.

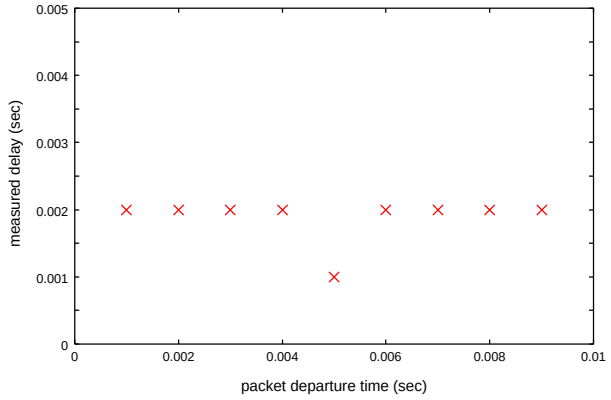

Fig. 8. Sample measured delay sequence

We now focus on the performance evaluation of the proposed available bandwidth estimation scheme. Fig. 9 shows the accuracy of the proposed available bandwidth estimation scheme for various target links. We observe that in most cases the real available bandwidth, *ABW ratio* in the figure, lies within σ from the average (μ) of the estimation values. We also observe that the averages of the estimation values are almost always above the real available bandwidth ratio. Although $F_{Q_{0,i+1}}(x)/F_{Q_{0,i}}(x)$ approaches to the available bandwidth ratio $(1-\rho_i)$ by (25) as x decreases to 0, x can not be zero due to the measurement resolution Δ and x may be a little larger than Δ as shown in (40) and (41). If x is positive, then $F_{Q_{0,i+1}}(x)/F_{Q_{0,i}}(x)$ is likely to be larger than $\omega_i (=1-\rho_i)$ since $R(x)$ is non-negative in (24).

Currently, about 3.8×10^5 packets are sent during one probing period since the probing rate and the probing duration are fixed to r_m and v_m. However, in the case of the second link between *CR1* and *CR2*, we need not send that many packets. If the utilization of the first and second links is 0.3, then according to (44) 9.6×10^4 packets need to be sent. Furthermore, since M_{th} needs not be achieved with j=1 for $M_{j\Delta}(k_{0,i})$ and $M_{j\Delta}(k_{0,i+1})$ by (41), the number of packets might be decreased further. Thus, we test the performance of the available bandwidth estimation scheme for the second link with a different number of probe packets. We change the number of packets with the probing rate while maintaining the probing duration at v_m. Fig. 10 shows that a reasonable performance is obtained even when the probing rate is as low as 100 Kbps. The real available bandwidth lies within σ from the average (μ) of the estimation values for every probing rate. We also observe that the variance of the estimation tends to decrease as the probing rate or the number of probe packets increases. This tendency is expected from (21). Figs. 9 and 10 show that the minimum required probing rate can be different depending on the hop count to the target link and the offered load. If the link utilizations are very low on some path, then it may be possible to probe links more than 10 hops away from the monitoring node with a limited number of probe packets by (44). We plan to investigate an adaptive probing rate-based available bandwidth estimation method in more detail in the future.

[3] We evaluate the detailed value of θ-quantile delay using linear interpolation if necessary.

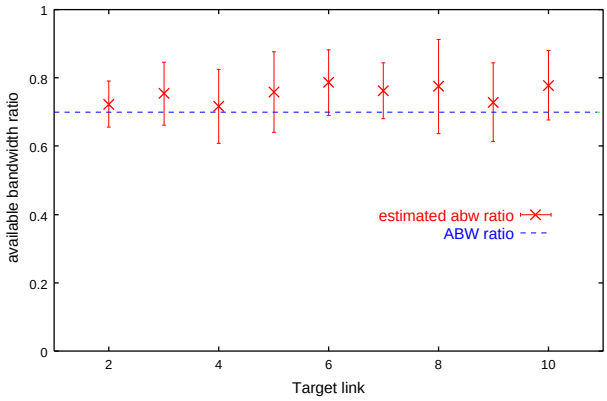

Fig. 9. Accuracy of the proposed available bandwidth estimation scheme for various target links

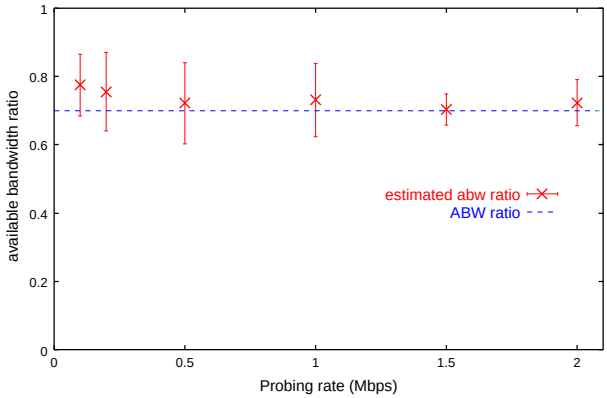

Fig. 10. Accuracy of the proposed available bandwidth estimation scheme for various probing rates

We evaluate the performance of the proposed available bandwidth estimation scheme in a more dynamic environment. The target link is the 7-th link between *CR6* and *CR7* in Fig. 6. The probing rate is 2.0 Mbps and the probing duration is set to 40 seconds. The offered load for the links other than the target link is about 0.3. Fig. 11 compares the estimated available bandwidth ratio with the measured available bandwidth ratio under a time-varying Poisson traffic load and Fig. 12 evaluates the performance of the proposed available bandwidth estimation scheme under a self-similar traffic load. Although there is a rather large error in the estimated value around 400 seconds in Fig. 11, these significant errors do not occur frequently and the estimation values follow the changes of the available bandwidth ratio most of the time. We also observe that the estimated available bandwidth ratio tracks significant changes of the measured available bandwidth for a self-similar traffic load as shown in Fig. 12.

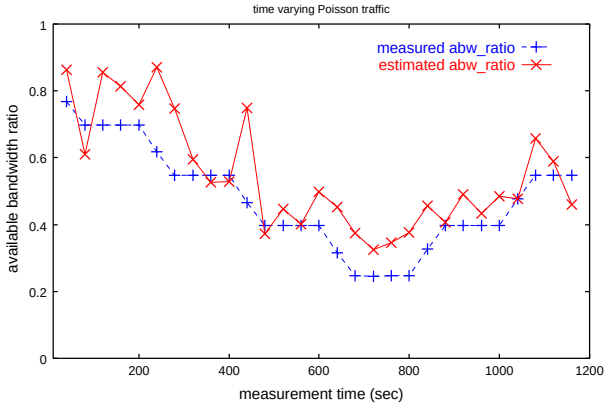

Fig. 11. Performance evaluation of the proposed available bandwidth estimation scheme under a time-varying Poisson traffic load

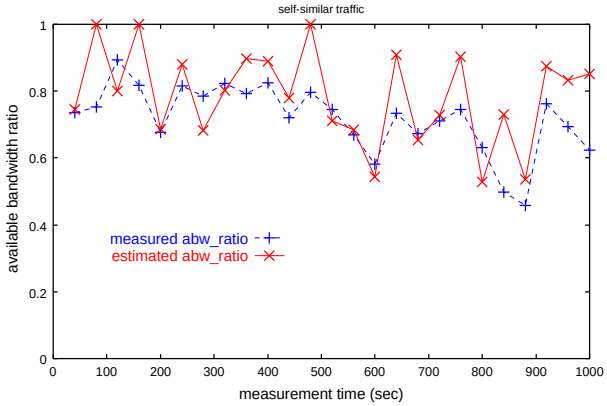

Fig. 12. Performance evaluation of the proposed available bandwidth estimation scheme under a self-similar traffic load

We finally evaluate the performance of the proposed available bandwidth estimation scheme for a link behind the tight link which has the minimum unused available bandwidth. In this case, the link rate of the link between *CR4* and *CR5* is changed to 150 Mbps and all other link rates remain at 1 Gbps. Since we apply 50 Mbps of traffic to the link *CR4-CR5*, the available bandwidth is only 100 Mbps and the *CR4-CR5* becomes the tight link. All the other conditions are almost the same as the case of Fig. 11. The target link is the 7-th link and a time-varying Poisson traffic load is offered to that link. Fig. 13 shows the test result for this case. The proposed scheme closely tracks the change of the available bandwidth although the target link is behind the tight link.

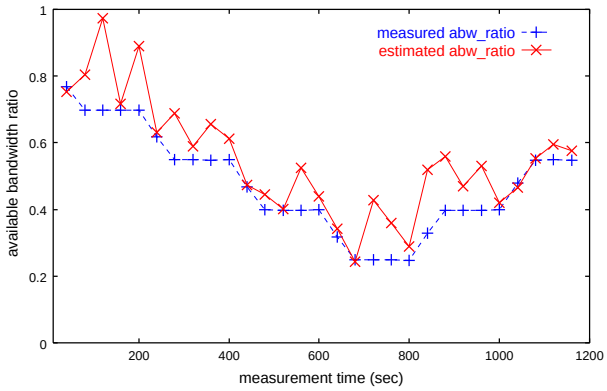

Fig. 13. Available bandwidth estimation for a link (7-th link) behind the tight link (5-th link) under a time-varying Poisson traffic load

6. Conclusions

In this paper, we proposed a scheme which estimates the available bandwidth ratio of a remote link or path segments without requiring remote nodes to deploy our tool. Since our scheme utilizes ICMP timestamp messages and 80% of routers respond to ICMP timestamp messages according to our measurement results, our scheme can be used to monitor many links or path segments around the monitoring node. We measure one-way delay from the difference of the packet sending time and the Timestamp value received from the remote node, extract the queueing delay component from the measured delay, and apply PASTA theory to estimate the product of the available bandwidth ratios of the links on a given path segment. Then, from the ratio of the available bandwidth ratio products we can infer the available bandwidth ratio of the target link. The use of ICMP timestamps entails two major challenges. One is the coarse resolution (1 msec) of timestamps and the other is the clock skew between different nodes. We first developed a statistical method to extract the queueing delay distribution from the coarse resolution delays. We also developed a new highly accurate clock skew rate estimation scheme. We evaluate the performance of the proposed available bandwidth estimation scheme through simulation and find that our scheme closely

estimates the available bandwidth ratio of remote links even when the target links are behind the tight link which has the minimum available bandwidth on a given path.

7. REFERENCES

[1] A. Yaar, A. Perrig, and D. Song, SIFF: a stateless internet flow filter to mitigate DDoS flooding attacks, In *Proc. IEEE Symposium on Security and Privacy*, May 2004.

[2] J. R. Crandall *et al*, On deriving unknown vulnerabilities from zeroday polymorphic and metamorphic worm exploits, In *Proc. ACM CCS*, Alexandria, Virginia, Nov. 2005.

[3] R. L. Carter and M. E. Crovella, Measuring bottleneck link speed in packet-switched networks, *Performance Evaluation*, 27-28, 1996.

[4] B. Melander, M. Bjorkman, and P. Gunningberg, A New End-to-End Probing and Analysis Method for Estimating Bandwidth Bottlenecks, In *Proc. IEEE GLOBECOM*, Nov. 2000.

[5] M. Jain and C. Dovrolis, Pathload: A measurement tool for end-to-end available bandwidth, In *Proc. PAM*, Mar. 2002.

[6] V. Ribeiro et al, pathChirp: Efficient available bandwidth estimation for network path, In *Proc. PAM*, Apr. 2003.

[7] J. Navratil and R. L. Cottrell, ABwE: A practical approach to available bandwidth estimation, In *Proc. PAM*, Apr. 2003.

[8] N. Hu and P. Steenkiste, Evaluation and Characterization of Available Bandwidth Probing Techniques, *IEEE JSAC*, 21(10), Aug. 2003.

[9] J. Strauss, D. Katabi, and F. Kaashoek, A measurement study of available bandwidth estimation tools, In *Proc. ACM IMC*, Florida, Oct. 2003.

[10] S. Y. Nam, S. Kim, J. Kim, and D. K. Sung, Probing-Based Estimation of End-to-End Available Bandwidth, *IEEE Communications Letters*, 8(10), June 2004.

[11] G. Jin, G. Yang, B. R. Crowley, and D. A. Agarwal, Network characterization service (NCS), Technical report, LBNL, 2001.

[12] A. Shriram et al., Comparison of public end-to-end bandwidth estimation tools on high-speed links, In *Proc. PAM*, Mar.-Apr. 2005.

[13] V. Paxson, On calibrating measurements of packet transit time, in *Proc. ACM SIGMETRICS*, pp. 11-21, June 1998.

[14] S. B. Moon, P. Skelly, and D. F. Towsley, Estimation and removal of clock skew from network delay measurements, in *Proc. IEEE INFOCOM*, Mar. 1999.

[15] L. Zhang, Z. Liu, and C. H. Xia, Clock synchronization algorithms for network measurements, in *Proc. IEEE INFOCOM*, pp. 160-169, June 2002.

[16] R. S. Ryger, fixclock: Removing clock artifacts from communication timestamps, Technical Report DCS/TR-1243, Yale University, March 2003.

[17] V. Paxson, End-to-end routing behavior in the internet, In *Proc. ACM SIGCOMM*, Aug. 1996.

[18] S. Chuang, A. Goel, N. McKeown, and B. Probhakar, Matching output queueing with a combined input output queued switch, In *Proc. IEEE INFOCOM*, Mar. 1999.

[19] H. Lee, and S. Seo, Matching output queueing with a multiple input/output-queued switch, In *Proc. IEEE INFOCOM*, Mar. 2004.

[20] R. W. Wolff, *Stochastic modeling and the theory of queues*, Prentice Hall, 1989.

[21] S. Karlin and H. M. Taylor, *A first course in stochastic processes, 2nd ed*., Academic Press, 1975.

[22] M. Harchol-Balter. *15-359 Probability and Computing – Inequalities*. http://www.cs.cmu.edu/~15359/notes/bounds.pdf, 2005.

[23] D. Mills, Network time protocol (version 3): Specification, implementation and analysis, Tech. Rep., Network Information Center, SRI International, Menlo Park, CA, March 1992.

[24] M. E. Crovella and A. Bestavros, Self-similarity in World Wide Web traffic: evidence and possible causes, *IEEE/ACM Trans. Networking*, 5(10), Dec. 1997.

[25] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, On the self-similar nature of Ethernet traffic, *IEEE/ACM Trans. Networking*, 2(1), Feb. 1994.

[26] V. Paxson and S. Floyd, Wide-area traffic: the failure of Poisson modeling, *IEEE/ACM Trans. Networking*, 3(3), June 1995.

[27] R. H. Riedi, M. S. Course, V. J. Ribeiro, and R. G. Baranuik, A multifractal wavelet model with application to network traffic, *IEEE Transactions on Information Theory*, 45(3), Apr. 1999.

[28] C. Fraleigh et al. Packet-level traffic measurements from the Sprint IP backbone. IEEE Network, 17(10):6-16, Nov.-Dec. 2003.