Estimation of the Available Bandwidth Ratio of a Remote Link or Path Segments

Seung Yeob Nam[a], Seong Joon Kim[b], Sihyung Lee[c], Hyong S. Kim[d]

[a]*Department of Information and Communication Engineering, Yeungnam University, Gyeongsan 712-749, Republic of Korea*
[b]*DMC Research Center, Samsung Electronics, Suwon 443-742, Republic of Korea*
[c]*Department of Computer and Information Security, Seoul Women's University, Seoul 139-774, Republic of Korea*
[d]*Dept. of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, United States*

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, the problem of estimating the available bandwidth for a specific remote link has not been investigated in detail yet. We propose a new scheme to estimate the available bandwidth ratio of a remote link or remote path segments, a group of consecutive links, without deploying our tool at the remote nodes. The scheme would be helpful in accurately pinpointing anomalous links. Two streams of ICMP timestamp packets are sent to both end nodes of a target link according to a Poisson process, and the available bandwidth ratio for the target link is estimated based on the measured packet delay. Since the proposed scheme needs not incur a short-term congestion, unlike conventional end-to-end available bandwidth estimation mechanisms, the intrusiveness is low and the proposed scheme overcomes the limitation of conventional approaches, inability to probe the links beyond the tight link with the minimum available bandwidth. The performance of the proposed scheme is evaluated by ns-2 simulation.

Keywords: Available bandwidth ratio, remote link probing, Poisson probing, ICMP timestamp

1. Introduction

The capacity of core networks has increased tremendously due to recent technology developments 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 attacks [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.

However, the Internet consists of many heterogeneous sub-networks, and the nodes belonging to different administrative domains are not easily accessible or visible to the network operators who do not own them. It is virtually impossible to deploy the probing tool in every router and thus, it is not easy to monitor every possible path or link with the conventional end-to-end available bandwidth estimation schemes. There were some approaches to infer the packet delays, packet reordering, and packet losses on the network links using tools deployed only on the sender side, e.g. tulip [3] and cing [4]. However, there has been no successful approach to the problem of estimating the available bandwidth of a remote link. If an end node, which can be a user node running TCP or peer-to-peer (P2P) application, a server belonging to a content delivery network (CDN), etc., knows the available bandwidth of each component link on a given path, then that information might be used to achieve lower latency or higher resource utilization by avoiding a set of congested links, or to perform detailed network diagnosis using correlation of bandwidth of adjacent links [5].

Thus, this paper considers the problem of estimating the available bandwidth ratio for a remote link or a path segment which consists of several consecutive links without deploying a monitoring program at the remote nodes. This problem has not yet been addressed extensively in the literature. Jin *et al.* [6] attempted to solve a similar problem. They developed *pipechar* to estimate the available bandwidth of each link on a given path. However, it has been reported that pipechar is unresponsive to variations in cross-traffic on 100 Mbps paths [7] and it has a fundamental limitation, the details of which are explained below.

There are several works on end-to-end available bandwidth estimation problem [8–18]. These techniques can be usually classified into two categories: probe gap model (PGM), and probe rate model (PRM) [13]. It is not easy to extend these techniques to estimate the available bandwidth of remote links. In particular, it is very difficult to probe the links beyond the tight link, which has the minimum unused bandwidth on a given path. The reason can be explained by the following example. Fig. 1 shows a path between two Nodes A and B. 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 the PGM-based 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. The PRM-based methods usually need to congest the target link temporarily by sending probe packets at a sufficiently high rate sometimes close to the link rate. But, we can easily know that the probing traffic may not induce congestion at any link after the tight link because of the lowest available bandwidth at the tight link.

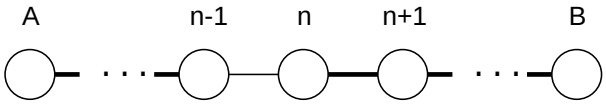


Figure 1: Sample network path

The closest to our objectives was Jin *et al.*'s pipechar [6]. Pipechar is also a 'Sender only' network probing program which aims to estimate the available bandwidth of each hop. However, pipechar throttles the network to each hop by sending a large burst of data through the link, which may cause further problems on the already congested network. Furthermore, since pipechar is also using packet trains to probe the network links [6], it also inherits the fundamental limitation of the PRM-based methods, i.e. pipechar can not estimate the available bandwidth of the links beyond the narrow link on a given path except some special case where a short-term congestion can be induced by applying cross traffic beyond the narrow link. However, finding a disjoint path that crosses only at or after the target link and applying cross traffic through that path are difficult problems in the current IP networks. Even though it is possible, the use of pipechar for the purpose of regular network monitoring is not recommended because of substantial traffic overhead [6].

Thus, the problem of estimating the available bandwidth of arbitrary remote links, especially the links beyond the narrow link or the tight link, has not been resolved using conventional approaches. As a first step to this problem, we propose a new technique to estimate the available bandwidth ratio of a remote link while overcoming the limitation of these conventional approaches. The available bandwidth ratio of a target link is the ratio of the available bandwidth to the link rate at the selected link. Since Harfoush *et al.*'s approach [19] can be used to estimate the link rate of a remote link, the available bandwidth might be estimated by combining our proposed mechanism and Harfoush's method [19]. However, we focus only on the estimation of available bandwidth ratio in this paper.

We first estimate the delay distribution for a path segment from the monitoring node to another remote node, and then from the obtained delay distribution we estimate the product of the available bandwidth ratio of each 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 the proposed scheme does not require any condition on the ratio of link rates of consecutive links, it can estimate the available bandwidth ratio of the links beyond the tight link on a given path without overloading any network link.

In order to measure the delay up to a remote node without deploying our program at remote nodes, we use ICMP timestamp messages in a similar way to tulip [3] and cing [4]. Since over 90% of routers respond to ICMP timestamp messages according to [3], we expect that our scheme can be used to monitor a wide range of links around the monitoring node. However, there are some 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 variations, 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, the 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 hence

the available bandwidth ratio significantly. There have been some approaches to address the clock skew problem [21–24], and among them we employ the linear algorithm-based approach [22] to correct the clock skew. Since the second issue can be resolved with existing approaches, we focus on the issue of coarse resolution of ICMP timestamps. We might consider the use of TCP timestamps provided through *TCP timestamps option* of TCP protocol instead of ICMP timestamps. However, the resolution of TCP timestamps can be significantly different depending on the OS type from 10 to 500 msec [25]. If the resolution of timestamps increases too high, then a reasonable accuracy may not be achieved with a limited amount of probe packets, which will be discussed in Sections 4 and 5. Thus, we consider only the use of ICMP timestamps in this paper.

The contribution of this paper can be summarized as follows.

- A mechanism for estimating the available bandwidth ratio or the product of available bandwidth ratios for a remote path segment is developed under accurate timing information, without deploying the probing tool at remote nodes.
- An available bandwidth ratio estimation mechanism which can probe a remote link or path segments even under coarse timestamp resolution of milliseconds is developed.
- The proposed scheme can estimate the available bandwidth ratio of a link even beyond the tight link on a given path.
- The proposed scheme is non-intrusive because it does not require even a short-term congestion on network links.

The rest of this paper is organized as follows. In Section 2, we discuss related works. In Section 3, we explain how to estimate the product of the available bandwidth ratios of the element links for a given path segment based on packet delays under the assumption that it is possible to measure the exact delay of each packet on a given path. In Section 4, we investigate the available bandwidth ratio estimation problem under the coarse resolution of ICMP timestamps, and briefly discuss the clock skew correction issue. In Section 5, we evaluate the proposed estimation scheme by simulation and finally, conclusions are given in Section 6.

2. Related Works

In our proposed available bandwidth ratio estimation scheme, accurate delay measurement or estimation is very important since this affects the accuracy of the method. The well-known tool ping can measure network round-trip times (RTTs) by sending ICMP requests to a target node over a short period of time. However, ping was not designed as a delay measurement tool, but it was a reachability testing tool. Using the same concept, the tool called pathchar [26] estimates the internal link delays. In order to estimate the delay of a remote link, this tool measures the round trip time to the head of the link and the round trip time to the tail of the link by eliciting TTL-expired responses from routers. The per-link delay is estimated by the difference of those two numbers divided by 2 assuming symmetry between the forward and the reverse paths. However, the route symmetry assumption is not generally valid [27]. Even when there is route symmetry, the delay on the forward path is usually not equal to that on the reverse path because the queueing delay component is normally different due to different cross traffic loads.

Thus, the schemes based on the route symmetry assumption cannot provide accurate one-way delay values.

There have been approaches to infer the internal network performance from end-to-end measurements. These inference techniques are usually referred to as network tomography and some of them have focused on deriving internal delay statistics by injecting probe packets from one source to multiple destinations and correlating the observed packet behavior on the resulting tree topology [28–30]. However, since the inference focuses on the statistics such as mean or variance, not on the individual packet behavior, this may not be an adequate approach to measure small queueing delays on each network link. Rizk and Fidler [31] attempted to identify the service curve for each link from the path-level measurements by applying network tomography techniques, and obtain the available bandwidth of each link from the derivative of the service curve. They used Legendre transform to change the min-plus convolution among the service curves of the links to the summation of the backlog curves of those links, and applied well-known tomography methods to obtain link-level service curves from the path-level measurements. TTL-based tailgaiting technique, i.e. dropping of loading packets around the target link, was used to obtain different path-level measurements. However, this tomography-based scheme also has a limitation that it cannot obtain the service curves for post-narrow links [31].

In order to overcome the limitations of ping and pathchar, tulip [3] and cing [4] estimate the per-link queueing delays using ICMP timestamp packets. If we send an ICMP timestamp request message to a responsive remote node, then the remote node will send back an ICMP timestamp reply message with three timing information in the fields of Originate timestamp, Receive timestamp, and Transmit timestamp. The Originate timestamp contains the time the sender last touched the message before sending it, the Receive timestamp contains the time the remote target node first touched the message on receipt, and the Transmit timestamp contains the time the remote node last touched the message before sending it. When it is necessary to measure the delay between Nodes n and $n + 1$ in Fig. 1, cing sends two back-to-back ICMP timestamp packets, one packet to Node n and the other packet to Node $n + 1$. If we assume that the two packets traverse the same path up to Node n and the two packets experience the same delay up to Node n, then we can estimate the delay between Nodes n and $n + 1$ from the difference of the two Receive timestamp values. cing and tulip use this idea to estimate the per-link delays. Our scheme follows the same approach, but our scheme does not use per-link delay estimates. In order to estimate the available bandwidth ratio based on the packet delays, it is important to know whether there is a non-zero queueing delay component from the delay measurements. When a 1500 byte packet is sent to a 1 Gbps link, the transmission time is only 12 usec and usually the queueing delay has the same order of magnitude under non-heavy traffic loads. Since the ICMP timestamps have a resolution of 1 msec, if we measure the per-link delay from the difference of two ICMP timestamps, the queueing delay components are hardly distinguishable. On the other hand, the packet sending times can be measured up to 1 usec resolution in common unix or linux machines, and thus, we attempt to estimate per-link available bandwidth based on the two end-to-end delays: one delay from the source node to Node n, and another delay from the source node to Node $n + 1$.

There is one more difficulty in using ICMP timestamps for accurate packet delay measurement in addition to the coarse resolution of timestamps. The problem is that Receive timestamp in the ICMP timestamp reply message does not indicate the accurate packet arrival time. Usually it takes time to generate ICMP reply messages [20] and the magnitude of ICMP generation delays can be larger than 1 ms. Routers usually insert just one timestamp, which is different from the actual packet arrival time, in both the Receive and Transmit timestamps when they gen-

erate the ICMP reply messages [3]. Thus, the value of the Receive timestamp is not obtained at the instant of ICMP packet arrival, but is obtained after some processing time component in the CPU. However, this processing time component is not explicitly characterized yet and can be different depending on the routers [4]. Thus, it is very difficult to separate the queueing delay and the processing time components from the delay calculated as the difference between the Receive timestamp and an accurate sending time[1]. Thus, we assume that the Receive timestamp value is written promptly on arrival of the ICMP timestamp request packet and this is assumed throughout this paper. Some special-purpose passive monitors timestamp each packet in the Network Interface Card (NIC) [34–36] to avoid these unexpected delays. Thus, we expect our scheme can be used in practical applications if this kind of NICs are partially deployed in the network, or ICMP timestamping is incorporated in the fast path of a router as suggested in [37]. But, path record fields and GPS time receiver are not required, and the timestamp need not be collected for every transit node in our case. These assumptions might be realized in the future internet environment, where the redesign of router architecture is also considered [38, 39].

Many schemes have been proposed to estimate the end-to-end available bandwidth [8–18]. Most of them can be classified into probe gap model (PGM) or packet rate model (PRM) [13], and they usually use a simplified model of single-hop path with fluid cross traffic even for multi-hop cases [40]. Liu *et al.* analyze the limitation of the fluid model in the single-hop environment [41] and in multi-hop environments [40], and show the gap between the real packet model and fluid model, i.e. the gap of response curves, can be mitigated by using large packet size or long packet-trains for packet train methods. Since the mechanism developed in this paper is neither PGM-based nor PRM-based method, those results are not directly related with our scheme.

There also have been some approaches [42, 43] to estimate the end-to-end available bandwidth in a "single-end" mode. One of the representative approaches is *abget* [42]. abget first sends a HTTP GET request to a TCP-based web server, and attempt to control the the downstream traffic rate by generating "fake ACKs", with appropriate ACK numbers and advertised window values. The principle of abget is similar to that of pathload [10]. Linkwidth [43] is another tool that estimates the end-to-end capacity and available bandwidth in a "single-end" mode. Linkwidth uses a modified version of recursive packet train (RPT), a technique originally used in Pathneck [44] to locate the tight link on a given path, to estimate the path capacity, and uses an extended version of Train of Packet-Pair (ToPP) algorithm [9] to estimate the end-to-end available bandwidth based on single-end controlled TCP packet probes. However, those mechanisms cannot be used to find the available bandwidth information for the post-narrow links due to the limitation of the PRM-based methods described in Section 1.

Baccelli *et al.* [45] investigated the role of PASTA especially in network delay measurements. They show that PASTA has a good property that Poisson sampling is unbiased even when observers become intrusive affecting the existing system, although Poisson probing may not be the best in terms of variance of the Mean Square Error (MSE). They also show that if non-intrusiveness is achieved through 'rare probing', e.g. by making the inter-arrival time large, the difference between the expected delay of the data packets in the unperturbed (or unprobed) system and the expected delay of the probes can be arbitrarily small also for Poisson probing. Since unbiased delay measurement is very important for available bandwidth estimation in our case, we use Poisson probing with a low probing rate.

[1]It is reported that ICMP packets usually have similar forwarding priorities as normal UDP or TCP packets that are processed through fast path when they are passing the router, not destined to it [32, 33]. We assume that the processing delay on the fast path is fixed.

Table 1: Major parameters and variables

N	Number of packets in a queueing system at an arbitrary time
C_i	Service rate of the output link of Node i
ρ_i	Offered load to the queue of Node i
L	Probe packet size
$Q_{0,n}$	Summation of queueing delays that a probe packet experiences at transit nodes from 0 to $n-1$
q_i	Queueing delay at Node i
$\tilde{s}_i$	Time interval from the arrival instant of the test packet p at Node i to the instant a timestamp is assigned to that packet
a_0	Packet sending time measured at the monitoring node
$a_i(p), i > 0$	Time (according to the clock of Node i) when a timestamp is assigned to the probe packet p by Node i
$D_{0,i}(p)$	Accurate delay of a probe packet p from Node 0 to i
$D_{m(0,i)}$	Minimum value of $D_{0,i}(p)$ for all p's
$a'_i(p), i > 0$	Timestamp value assigned to the packet p at time $a_i(p)$ by the clock of Node i
$D'_{0,i}(p)$	Delay between Nodes 0 and i measured from timestamp values
$D'_{m(0,i)}$	Minimum value of $D'_{0,i}(p)$ for all p's
Ω	Unit of ICMP timestamp measurement (Ω = 1 msec)
Δ	Time measurement resolution in usual linux machines
$M(k)$	Number of packets which experience the delay (measured from timestamps) within Δ from the minimum delay $D'_{m(0,i)}$ among the k sent packets
$M_{j\Delta}(k)$	Number of packets which experience the delay within $j\Delta$ from the minimum delay $D'_{m(0,i)}$ among the k sent packets
$k_{0,i}$	Number of probe packets sent from the monitoring node (Node 0) to Node i
r_m	Maximum limit on the probing rate
v_m	Maximum limit on the probing duration
α	Clock skew rate
α'	Estimated value of α

3. Basic Idea of Available Bandwidth Ratio Estimation

In this section, we explain how to estimate the ratio of the available bandwidth to the link rate, which is referred to as the available bandwidth ratio, based on the packet queueing delays. In order to estimate the packet queueing delays, we use ICMP packets. We begin this section with a set of assumptions and requirements in Subsection 3.1. Then, we investigate how to estimate the available bandwidth ratio of a single link in Subsection 3.2. In Subsection 3.3, we consider the problem of estimating the product of the available bandwidth ratios of the links constituting a target path segment. Throughout this section, we assume that ICMP packets provide accurate timing information and there is no clock skew between any two different nodes. We relax these assumptions in Section 4. Table 1 summarizes the major parameters and variables.

3.1. 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 the end nodes should coincide up to the closer end node.

We assume that the route from the monitoring node to both the end nodes does not change during one probing period [46]. 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, such as traceroute.

3.2. 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 serve the arriving packets in exactly the same order as the output queued switches [47, 48]. 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 the test traffic and the 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. According to [49, Theorem 6 in Chap. 5], PASTA (Poisson-Arrival See Time Averages) holds under Lack of Anticipation Assumption (LAA). Because a Poisson process has independent increments, we expect LAA will be valid in the usual network environment and the validity of PASTA have been shown for various traffic patterns in [45]. Thus, using PASTA [49], 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 the 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 for the one hop case can be decomposed into four components: processing delay, queueing delay, transmission delay, and propagation delay. If we fix the size of the test packets to L, then the transmission delay is fixed to L/C. The propagation delay and the processing delay (on the fast path in router processing) 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}) for 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.

3.3. Multiple Hop Case

Although it is reported that over 90% of routers respond to ICMP timestamp packets [3], there might be some cases where the two requirements in Subsection 3.1 are not valid for the link of interest. In that case, the available bandwidth of the specific link may not be inferred. But, if we find two nodes which embrace the target link and satisfy the two requirements of Subsection 3.1, we can monitor the behavior of the links between the selected two nodes aggregately using the approaches described here[2]. Hereafter, 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 $n+m$ respond to ICMP timestamp messages and the path from the monitoring node (Node 0) to Node n coincides 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 from the monitoring node to Node n. We send a group of probe packets to Node n according to a Poisson process. Let $a_0(p)$ be the time when the probe packet p is sent from the monitoring node. Let $a_n(p)$ be the value of ICMP timestamp which is assigned at the instant the probe packet p arrives at Node n. We define $a_{min}(0,n)$ as

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

[2]The two embracing nodes can be searched using the concept of tomography group described in [4].

$a_n(p) - a_0(p)$ 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. We assume that the tandem nodes on a selected path can be modeled as a Jackson queueing network that satisfies the following properties:

- The cross traffic from the nodes outside of Jackson network arrives according to a Poisson process.
- The number of the server at each node is one, and the service time is exponentially distributed.
- A packet being served by Node i can be delivered to the next hop node $(i+1)$ on the path with a probability of $r_{i\{i+1\}}$ or delivered to other nodes not on the considered path with a probability of $1 - r_{i\{i+1\}}$.

The third property might be approximately valid in the internet. The number of flows passing core routers is usually very high up to million. If such a large number of flows are multiplexed in core routers, each core router may feel that the next hop of each packet is determined randomly. Walrand and Varaiya [50] have shown that the sojourn times of a packet at the various nodes of a non-overtaking path are all mutually independent in any open Jacksonian network. Since we assume a single server with First-Come, First-Served (FCFS) service policy at each node and we do not consider the re-entry of a packet that has left the path already, the non-overtaking path condition is valid in our case. Based on the result of [50], we assume that N_i^-'s are mutually independent, and we obtain

$$\Pr(N_0^- = 0, N_1^- = 0, \cdots, N_{n-1}^- = 0) = \Pr(N_0^- = 0) \Pr(N_1^- = 0) \cdots \Pr(N_{n-1}^- = 0). \tag{4}$$

The following relation is valid by the arrival theorem for Jackson network [51]:

$$\Pr(N_i^- = 0) = \Pr(N_i = 0), \qquad i = 0, \ldots, n-1. \tag{5}$$

Combining (1), (4), and (5) yields

$$\Pr(N_0^- = 0, N_1^- = 0, \cdots, N_{n-1}^- = 0) = (1-\rho_0)(1-\rho_1)\cdots(1-\rho_{n-1}). \tag{6}$$

where ρ_i is the offered load to the queue of Node i. (6) is derived for Jackson network. However, it is not easy to derive such a simple form of relation for the queueing network where more realistic long-range dependent traffic arrives. Thus, we use (6) to resolve our problem assuming that the relation will be asymptotically valid even for non-Poisson traffic patterns. The validity of (6) is investigated under non-Poisson traffic indirectly based on the accuracy of the proposed available bandwidth ratio estimation mechanism evaluated through simulation in Section 5.

A random variable $Q_{0,n}$ denotes the summation of the 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 $j = 0, 1, \ldots, n-1$. Thus, we have

$$\Pr(Q_{0,n} = 0) = \Pr(N_0^- = 0, N_1^- = 0, \cdots, N_{n-1}^- = 0). \tag{7}$$

Combining (6) and (7) yields

$$\Pr(Q_{0,n} = 0) = (1 - \rho_0)(1 - \rho_1) \cdots (1 - \rho_{n-1}). \tag{8}$$

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

$$\Pr(Q_{0,n+m} = 0) = (1 - \rho_0)(1 - \rho_1) \cdots (1 - \rho_{n+m-1}). \tag{9}$$

From (8) and (9), we can use the following statistic to estimate $(1 - \rho_n) \cdots (1 - \rho_{n+m-1})$:

$$a(n, n + m) = \frac{\Pr(Q_{0,n+m} = 0)}{\Pr(Q_{0,n} = 0)}. \tag{10}$$

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 (10).

4. Estimation of Available Bandwidth Ratio 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 the remote node accurately. Usually for linux or unix machines, it is possible to measure the 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 the 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 the ICMP timestamps. We first assume that there is no clock skew problem between different nodes, and briefly state how we address the clock skew issue later in this section.

Although the resolution of ICMP timestamps is coarse as a millisecond, since the sending time can be measured down to microseconds at the probing node, 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 the ICMP probe packets.

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

$$D_{0,i}(p) = a_i(p) - a_0(p). \tag{11}$$

When $i > 0$, we cannot know the accurate value of $a_i(p)$ due to the coarse resolution of the 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 time $a_i(p)$ by the clock of 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

$$D'_{0,i}(p) = a'_i(p) - a_0(p), \tag{12}$$

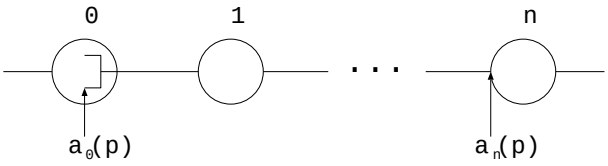


Figure 2: Parameters related with the delay of the probe packet p

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

We next investigate the accurate delay $D_{0,i}(p)$ in more detail for estimation of the available bandwidth ratio. 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 denote the transmission delay, the queueing delay, and the 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 receives a timestamp at Node i, $D_{0,i}(p)$ can be expressed as

$$D_{0,i}(p) = \sum_{m=0}^{i-1} \left\{ s_m^p + q_m^p + t_m^p + g_m \right\} + \tilde{s}_i,$$

where $\tilde{s}_i$ is the time interval from the arrival instant of the test packet p at Node i to the instant a timestamp is assigned to that packet. 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 m is fixed to s_m. When we fix the test packet size to L, we have

$$D_{0,i}(p) = D_{0,i}^f + Q_{0,i}(p) + \tilde{s}_i, \tag{13}$$

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 summation of variable components $Q_{0,i}(p)$ and $\tilde{s}_i$. We estimate the probability $\Pr(Q_{0,i} \leq 0)$ in order to know the product of the available bandwidth ratios according to (8).

We now investigate a relation among the measurable delay $D'_{0,i}(p)$, the queueing delay $Q_{0,i}(p)$, and the ICMP processing time $\tilde{s}_i$. Fig. 3 shows the 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

$$\xi = a_i(p) - a'_i(p), \tag{14}$$

then $0 \leq \xi < \Omega$. 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. The proof is given in [52, Proposition 1]. □

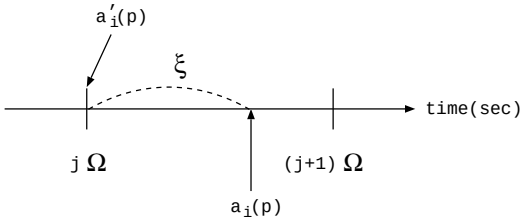


Figure 3: Relation between $a_i(p)$ and $a'_i(p)$ (j is the largest integer less than or equal to $a_i(p)/\Omega$)

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 (13) we have

$$D_{m(0,i)} = D^f_{0,i},$$

when $Q_{0,i}(p) = 0$ and $\tilde{s}_i = 0$. If we put $D'_{m(0,i)} = \min_p D'_{0,i}(p)$, from (12) and (14) we have $D'_{m(0,i)} = \min_p \{a_i(p) - a_0(p) - \xi\}$. Since $a_i(p) - a_0(p) = D_{0,i}(p)$ by (11), we have

$$\begin{aligned} D'_{m(0,i)} &= \min_p \{D_{0,i}(p) - \xi\} \\ &\geq \min_p \{D_{0,i}(p)\} - \max_p \{\xi\} \\ &> D^f_{0,i} - \Omega. \end{aligned}$$

Although $D^f_{0,i} - \Omega$ is a lower bound of $D'_{m(0,i)}$ by the above inequality, there may exist some packet p for which 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 $D^f_{0,i} - \Omega$. Since $D'_{0,i}(p) = D^f_{0,i} + Q_{0,i}(p) + \tilde{s}_i - \xi$ by (11), (12), (13) and (14), we have

$$\Pr(D'_{0,i} - D'_{m(0,i)} \leq x) = \Pr(Q_{0,i} + \tilde{s}_i - \xi + \Omega \leq x), \tag{15}$$

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 (15) can be rewritten as

$$\Pr(D'_{0,i} - D'_{m(0,i)} \leq x) = \Pr(Q_{0,i} + \tilde{s}_i + \gamma \leq x). \tag{16}$$

Since ξ is determined by the relative position of arrival time $a_i(p)$ in a time window of length Ω as shown in Fig. 3, we assume that ξ and γ ($\gamma = \Omega - \xi$) are independent of the queueing delay $Q_{0,i}$ and the ICMP processing time $\tilde{s}_i$. We also consider it reasonable to assume that $Q_{0,i}$ and $\tilde{s}_i$ are mutually independent from each other since $Q_{0,i}$ is the queueing behavior up to Node $i-1$ and $\tilde{s}_i$ is the ICMP packet processing time at Node i. Under these mutual independence assumption among $Q_{0,i}$, $\tilde{s}_i$, and ξ, the distribution of the measured delay $D'_{0,i} - D'_{m(0,i)}$ becomes equal to the convolution of the cdf of the queueing delay $Q_{0,i}$, a probability density function (pdf) of $\tilde{s}_i$ and a uniform pdf of γ in (16). Let $F_D(x)$ and $F_Q(x)$ denote the cumulative distribution functions of $D'_{0,i} - D'_{m(0,i)}$ and $Q_{0,i}$, respectively, i.e. $F_D(x) = \Pr(D'_{0,i} - D'_{m(0,i)} \leq x)$ and $F_Q(x) = \Pr(Q_{0,i} \leq x)$. Let $f_{\tilde{s}}(x)$ and $f_\gamma(x)$ denote the pdfs of $\tilde{s}_i$ and γ, respectively. Let $G_D(s)$, $G_Q(s)$, $G_{\tilde{s}}(s)$, and $G_\gamma(s)$ denote the Laplace transforms of $F_D(x)$, $F_Q(x)$, $f_{\tilde{s}}(x)$, and $f_\gamma(x)$, respectively. Then, under the mutual independence assumption among $Q_{0,i}$, $\tilde{s}_i$, and ξ, taking the Laplace transform on both sides of (16) gives

$$G_D(s) = G_Q(s)G_{\tilde{s}}(s)G_\gamma(s).$$

The distribution of $D'_{0,i} - D'_{m(0,i)}$ can be obtained from the measured delay and γ is approximately uniformly distributed over the interval $(0, \Omega]$ by Proposition 1. Thus, if we know the distribution of $\tilde{s}_i$, ICMP processing time at Node i, then we can calculate the distribution of $Q_{0,i}$ by taking the inverse Laplace transform on $G_D(s)/\{G_{\tilde{s}}(s)G_\gamma(s)\}$, i.e.

$$\mathcal{L}^{-1}\left\{G_D(s)/\{G_{\tilde{s}}(s)G_\gamma(s)\}\right\} = \mathcal{L}^{-1}\{G_Q(s)\} = F_Q(x) = \Pr(Q_{0,i} \le x),$$

where $\mathcal{L}^{-1}$ represents inverse Laplace transform. However, it is not easy to know the distribution of ICMP processing time at remote nodes because the characteristics of ICMP processing time can be different depending on the machines [4, 21]. We briefly investigate the effect of unknown ICMP processing time on the accuracy of queueing delay distribution ($\Pr(Q_{0,i} \le x)$) estimation. If the distribution of the ICMP processing time is unknown, we can estimate the queueing delay distribution in the following way:

$$\mathcal{L}^{-1}\left\{G_D(s)/G_\gamma(s)\right\} = \mathcal{L}^{-1}\left\{G_Q(s)G_{\tilde{s}}(s)\right\} = F_Q(x) * f_\gamma(x).$$

Then, we can define the estimation error as

$$\begin{aligned} error(x) &= \frac{\Pr(Q_{0,i} \le x) - F_Q(x) * f_{\tilde{s}}(x)}{\Pr(Q_{0,i} \le x)} \\ &= 1 - \frac{\int_0^x \Pr(Q_{0,i} \le x - \tau) f_{\tilde{s}}(\tau) d\tau}{\Pr(Q_{0,i} \le x)}. \end{aligned}$$

Since $\Pr(Q_{0,i} \le 0) \le \Pr(Q_{0,i} \le x - \tau) \le \Pr(Q_{0,i} \le x)$ for $0 \le \tau \le x$, we can derive the following lower and upper bound of $error(x)$:

$$1 - \int_0^x f_{\tilde{s}}(\tau)d\tau \le error(x) \le 1 - \frac{\Pr(Q_{0,i} \le 0)}{\Pr(Q_{0,i} \le x)} \int_0^x f_{\tilde{s}}(\tau)d\tau. \tag{17}$$

$error(x)$ is the relative error that results from the remaining unknown component of ICMP processing time. Let us focus on the upper bound. We analyze the queueing distribution to estimate $\Pr(Q_{0,i} \le 0)$. Thus, we are interested in the behavior of $\Pr(Q_{0,i} \le x)$ for very small values of x, and for those values of x the ratio $\Pr(Q_{0,i} \le 0)/\Pr(Q_{0,i} \le x)$ can be close to 1. If the value of $\int_0^x f_{\tilde{s}}(\tau)d\tau$ is close to 1, then $error(x)$ can be kept small by (17). Let us consider the lower bound of (17). We find that if the value of $\int_0^x f_{\tilde{s}}(\tau)d\tau$ is very small, the relative error can increase up to near 1. Thus, in order to maintain small values of the relative error, the pdf of the ICMP processing time needs to be clustered near $x = 0$. Even though ICMP processing time is not zero, if it is constant or densely clustered near some fixed value, then the fixed component can be captured by $D^f_{0,i}$ in (13) and $D'_{m(0,i)}$, and we can obtain the same upper bound as the case with zero or near zero ICMP processing times.

Hereafter, we assume that the ICMP processing time at the target node i, $\tilde{s}_i$, is zero, and we focus on recovering the distribution of $Q_{0,i}$ based on the measured data without resorting to inverse Laplace transform. Since $Q_{0,i}$ is assumed to be independent of γ ($\gamma = \Omega - \xi$), (16) can be expressed as

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

Since we consider the issues resulting from the coarse resolution of ICMP timestamps, we assume $\Omega \gg \Delta$ in this section. For small values of x less than Ω, the above relation can be changed into

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

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 $x = j\Delta$, the above equation can be expressed as

$$\Pr(D'_{0,i} - D'_{m(0,i)} \le j\Delta) = \frac{1}{\Omega} \sum_{k=0}^{j-1} \int_{k\Delta}^{(k+1)\Delta} \Pr(Q_{0,i} \le t)dt, \quad j < \Omega/\Delta. \tag{18}$$

We assume that in a very short interval of $[k\Delta, (k+1)\Delta]$, $\Pr(Q_{0,i} \le t)$ can be linearly approximated as $\Pr(Q_{0,i} \le t) \approx \alpha_k t + \beta_k$. From (18), and the piecewise linear assumption for $\Pr(Q_{0,i} \le t)$, we obtain the following relations. The detailed derivation is given in Appendix A.

$$\begin{aligned} \Pr(Q_{0,i} \le \frac{\Delta}{2}) &\approx \frac{\Omega}{\Delta} \Pr(D'_{0,i} - D'_{m(0,i)} \le \Delta), \\ \Pr(Q_{0,i} \le (n-\frac{1}{2})\Delta) &\approx \frac{\Omega}{\Delta} \left\{ \Pr(D'_{0,i} - D'_{m(0,i)} \le n\Delta) - \Pr(D'_{0,i} - D'_{m(0,i)} \le (n-1)\Delta) \right\}, \ 2 \le n < \Omega/\Delta. \end{aligned} \tag{19}$$

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

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 (8), or the product of the available bandwidth ratios for a path segment, starting from a remote node, by (10). According to (8) or (10), 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. Even though the probability $\Pr(Q_{0,i} \le \Delta/2)$ is close to 1, $\Pr(D'_{0,i} - D'_{m(0,i)} \le \Delta) \approx \Delta/\Omega = 0.001$ by (20). 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 occurrences 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. We see that if the

number of packet samples is not enough, the estimation error can be significantly large. Thus, we can easily know 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, especially the TCP flows, and a long probing time may hinder real-time monitoring of the available bandwidth ratio, the number of probe packets needs to be limited in real applications. Even though we send a sufficient number of packets considering the average load, if the load on the target path segment increases significantly, $M(k)$ might become 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, which is not starting from the monitoring node. We also assume that the target path segment consists of a single link for convenience and ease of providing the explanations. $k_{0,i}$ denotes the number of the 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

$$\hat{a}(i, i+1) = \frac{M_\Delta(k_{0,i+1})/k_{0,i+1}}{M_\Delta(k_{0,i})/k_{0,i}} \tag{21}$$

based on (10) and (20) under the assumption that $\Pr(Q_{0,i} \le 0) \approx \Pr(Q_{0,i} \le \Delta/2)$.

By the definition of $Q_{0,i}$ in (13), $Q_{0,i+1} = Q_{0,i} + q_i$, 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

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

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

$$F_{Q_{0,i+1}}(x) = \int_0^x F_{Q_{0,i}}(x - y) f_{q_i}(y) dy, \tag{22}$$

If we separate zero queueing delay component and non-zero queueing delay component in the pdf of q_i, then $f_{q_i}(y)$ can be modeled as

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

where $\delta(y)$ is the Dirac delta function, and the function $\tilde{f}_{q_i}(y)$ corresponding to the non-zero queueing delay component is assumed to be bounded. Then, the available bandwidth ratio of the link between Nodes i and $i+1$ is $\Pr(q_i \le 0) = \omega_i$ since $\Pr(q_i \le 0) = 1 - \rho_i$, and (22) can be changed into

$$F_{Q_{0,i+1}}(x) = \omega_i F_{Q_{0,i}}(x) + R(x), \tag{23}$$

where $R(x) = (1 - \omega_i) \int_0^x F_{Q_{0,i}}(x - y) \tilde{f}_{q_i}(y) dy$. Since $R(x) \le (1 - \omega_i) F_{Q_{0,i}}(x) \int_0^x \tilde{f}_{q_i}(y) dy$ and $\int_0^x \tilde{f}_{q_i}(y) dy \le x \cdot \max_{0 \le y \le x} \tilde{f}_{q_i}(y)$, $\lim_{x \to 0} R(x) = 0$ and from (23) we can obtain

$$\lim_{x \to 0} F_{Q_{0,i+1}}(x) / F_{Q_{0,i}}(x) = \omega_i = \Pr(q_i \le 0). \tag{24}$$

By (24), 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

$$F_{Q_{0,i+1}}(x)/F_{Q_{0,i}}(x) = \Pr(Q_{0,i+1} \le x)/\Pr(Q_{0,i} \le x). \tag{25}$$

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 (21), then we can use the statistic

$$\tilde{a}(i, i+1) = \frac{M_{j\Delta}(k_{0,i+1})/k_{0,i+1}}{M_{j\Delta}(k_{0,i})/k_{0,i}} \tag{26}$$

to estimate the available bandwidth ratio of the link between Nodes i and $i+1$ based on (25). In (25), 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 (25) and (26) also approach 1. Thus, the value of j needs to be kept as small as possible in (26).

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.

$$f_{q_i}(y) = \omega_i \delta(y) + (1-\omega_i)\lambda e^{-\lambda y}, \tag{27}$$

From (23) and the definition of $R(x)$, we can obtain

$$F_{Q_{0,i+1}}(x)/F_{Q_{0,i}}(x) \le \omega_i + (1-\omega_i)(1-e^{-\lambda x}). \tag{28}$$

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

$$x < -\frac{1}{\lambda} \ln\left(1 - \frac{\zeta}{1-\omega_i}\right).$$

Since the expectation of the queueing delay q_i is given as $\mu_{q_i} = E[q_i] = (1-\omega_i)/\lambda$ from (27), the above inequality can be changed into

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

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 (26), increases as the average queueing delay μ_{q_i} increases consistently as our intuition.

Thus far, we assumed that there is no clock skew between the monitoring node and a remote node. In reality, clock skew exists between different machines and it is important to accurately estimate the clock skew in order to remove or minimize the effect of clock skew on the queueing delay distribution.

In order to define clock skew in a formal way, we use the time at the monitoring node as the reference time t. Let $z(t)$ denote the value of the clock at the remote node at time t of 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*.

We use the well-known Moon *et al.*'s linear programming-based approach [22] to estimate the clock skew rate α. If the clock skew rate α is estimated, then we get rid of the component due to the clock skew rate from the measured delay by [52, eq. (30)] and we apply the available bandwidth estimation scheme developed earlier in this section to the skew-corrected delay sequence.

5. Numerical Results

In this section, we evaluate the performance of the proposed available bandwidth ratio estimation scheme under the fine-grained timestamp and the coarse-grained timestamp environments through ns-2 simulation. The fine-grained timestamp environment corresponds to the case where accurate probe packet arrival times are provided by remote hosts, and the coarse-grained timestamp environment corresponds to the case where coarse-grained packet arrival times are provided by remote hosts in units of Ω, i.e. msec. We assume that the packet sending times are measured accurately at the sending node. The evaluation results can be summarized as follows:

- We correct the clock skew using Moon *et al.*'s linear programming-based method [22] and find that this method can correct the clock skew with an error close to zero. But, the results are not included due to space limitation. Please refer to [52, Sec. 5] for the detailed results.
- We evaluate our proposed method with self-similar traffic as well as TCP traffic loads. We demonstrate that our proposed method closely estimates the available bandwidth ratio even with a coarse-grained timestamp, i.e. the measured available bandwidth ratio falls within the standard deviation from the average of the estimated available bandwidth.
- We show some cases where we can reduce the probing rate of our scheme from 2 Mbps to a hundred Kbps without sacrificing the estimation accuracy significantly especially for close links.
- Finally, we demonstrate that our method can closely estimate the available bandwidth ratio of a link which is even behind a tight link on the same path. Such estimation is not possible with current conventional approaches.

Fig. 4 shows the network topology for ns-2 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 CR*i* and ER1 are responding to ICMP timestamp packets. The source node S*i* sends cross traffic to the destination node D*i* 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 three types of traffic patterns, Poisson, self-similar, and TCP 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 [54–56]. We use a multi-fractal model [57] to generate the self-similar traffic pattern and the Hurst parameter is set to 0.8.

The size of each packet of the cross traffic is selected from the following distribution: 40 bytes - 60%, 576 bytes - 20%, 1500 bytes - 20%.

We already noted that the estimator for the available bandwidth ratio (21) 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. $k_{0,i} = k_{0,i+1}$, (21) is simplified to $\hat{a}(i, i{+}1) = M_\Delta(k_{0,i+1})/M_\Delta(k_{0,i})$

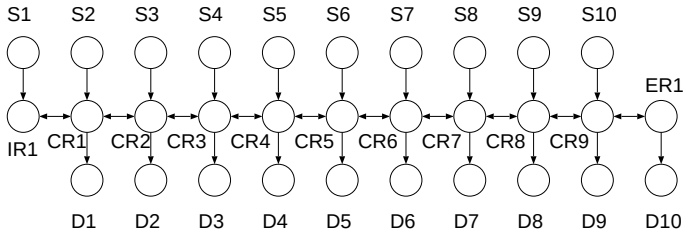


Figure 4: Simulation network topology

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 have suggested another estimator $\tilde{a}(i, i+1)$ in (26). When $k_{0,i} = k_{0,i+1}$, we can obtain a similar form as

$$\tilde{a}(i, i+1) = \frac{M_{j\Delta}(k_{0,i+1})}{M_{j\Delta}(k_{0,i})}. \tag{30}$$

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

$$j = \min\left\{j' | M_{j'\Delta}(k_{0,i+1}) \geq M_{th}, M_{j'\Delta}(k_{0,i}) \geq M_{th}\right\}, \tag{31}$$

where M_{th} is a threshold used to determine an appropriate minimum value of the numerator or denominator of (30). 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 it 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

$$\Pr(D'_{0,j} - D'_{m(0,j)} \leq \Delta) \approx \frac{\Delta}{\Omega} \Pr(Q_{0,i} \leq 0). \tag{32}$$

Combining (8) and (32) yields

$$\Pr(D'_{0,j} - D'_{m(0,j)} \leq \Delta) \approx \frac{\Delta}{\Omega}(1-\rho_1)(1-\rho_2)\cdots(1-\rho_{i-1}). \tag{33}$$

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

$$\frac{M_\Delta(k_{0,i})}{k_{0,i}} \approx \frac{\Delta}{\Omega}(1-\rho_1)(1-\rho_2)\cdots(1-\rho_{i-1}). \tag{34}$$

If we want to send enough probe packets so that j can be 1 in (31), i.e. $M_\Delta(k_{0,i}) \geq M_{th}$, then a lower bound of $k_{0,i}$ can be derived from (34) as

$$k_{0,i} \geq \frac{\Omega}{\Delta} \frac{M_{th}}{(1-\rho_1)(1-\rho_2)\cdots(1-\rho_{i-1})}. \tag{35}$$

From the above inequality, we can easily establish that more packets need to be sent to probe farther links. However, the probing rate needs to be limited in order to prevent degradation of the data traffic performance due to probe traffic. In addition, the probing duration also needs 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

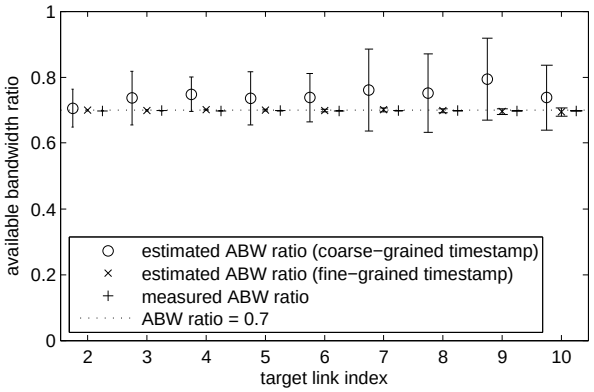
$$k_{0,i} \leq \frac{r_m v_m}{L}, \tag{36}$$

where L is the probe packet size and L is fixed to 40 bytes. (35) and (36) imply when $k_{0,i}$ is limited, the scope of the probing scheme, i.e. i in (35), is likely to be limited. However, it is reported that the link utilizations are usually less than 50% because of bandwidth over-provisioning in the core network [36]. Thus, as long as the link utilizations stay low, our scheme may be used to probe the links that are a 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 of the target link, are occupying only 0.4% of the 1 Gbps link and we assume that the effect of the probing traffic load on the performance of data traffic is negligible. The probing rate and the probing duration are set to r_m and v_m, respectively, if not specified otherwise. Since the probe packet size is fixed at 40 bytes, about 3.8×10^5 packets are sent during one probing period. In Fig. 4, the clock at IR1 is the reference clock and the clock skew rate (α) of Node CRi is set to $(i + 1) \times 5.0 \times 10^{-7}$.

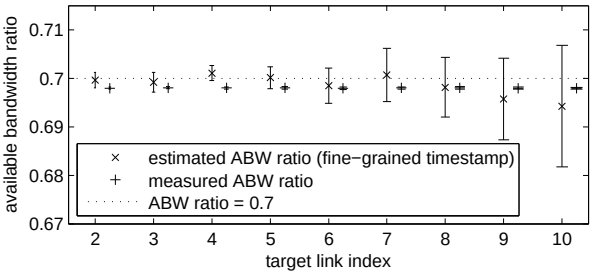
Fig. 5 shows the accuracy of the proposed available bandwidth ratio estimation scheme for various target links under Poisson traffic loads. Especially, Fig. 5(a) shows the estimation results under fine-grained and coarse-grained timestamp environments simultaneously and Fig. 5(b) shows the magnified version of the estimation results under fine-grained timestamps. When the fine-grained timestamps are provided by remote nodes, we use the mechanism developed in Section 3, i.e. (10), to estimate the available bandwidth ratio. When coarse-grained timestamps are provided, we use the mechanism developed in Section 4, i.e. (26), to estimate the available bandwidth ratios of the remote links. In Fig. 5(b) we find that the measured value of the available bandwidth ratio lies within the standard deviation (σ) from the average (μ) of the estimation values in most links. Although the variance increases as we probe farther links, we find that the standard deviation σ is still very small compared with the average μ in Fig. 5(a) especially when we estimate the available bandwidth ratio using fine-grained timestamps. Although the offered traffic load is 0.3 in Fig. 5(b), the measured available bandwidth ratio is slightly lower than 0.7. The difference corresponds to the load of the probe traffic. For example, when we probe the i-th link between Node $(i-1)$ and Node i, we send two probing streams with the traffic rate of 2 Mbps each, one stream to Node $(i - 1)$ and another stream to Node i. Then, the i-th link carries one probe packet stream at the rate of 2 Mbps during the probing period, and the available bandwidth ratio for the i-th link decreases by 0.002, i.e. 2 Mbps/1 Gbps, which is intrusiveness of our probing scheme. But, this intrusiveness could be lowered by decreasing the probing rate possibly at the cost of an increased probing duration. In Fig. 5(a) when coarse-grained timestamps are used, we observe that the measured available bandwidth ratio lies within σ from the average (μ) of the estimated values in most cases.

Figs. 6 and 7 show the performance of the proposed scheme for various target links under self-similar traffic and aggregated TCP (Reno) traffic loads, respectively. The number of TCP connections for each link is chosen to generate a traffic load of about 0.3. Although the variance of estimation increases compared to the result under Poisson traffic loads, the proposed scheme still works well.

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



(a) Comparison of the available bandwidth ratio estimation results under fine-grained and coarse-grained (ICMP) timestamps



(b) Magnified version of the available bandwidth ratio estimation results under fine-grained timestamps

Figure 5: Accuracy of the proposed available bandwidth ratio estimation scheme for various target links under Poisson traffic loads

links is 0.3, then according to (35), 9.6×10^4 packets need to be sent. Furthermore, since M_{th} does not need to be achieved with $j = 1$ for $M_{j\Delta}(k_{0,i})$ and $M_{j\Delta}(k_{0,i+1})$ in (31), 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. In Fig. 8 we find that very accurate estimation results are obtained even at the probing rate of 100 Kbps especially when fine-grained timestamps are replied by the remote nodes. Even though coarse-grained timestamps, with a resolution of 1 msec, are provided by the remote nodes, a reasonable performance is obtained for the probing rate of as low as 100 Kbps. The real available bandwidth ratio lies within σ from the average (μ) of the estimation values for most probing rates. We obtained similar results for self-similar and TCP traffic loads. We also observe that the variance of the estimation tends to decrease, and the accuracy of the estimation scheme based on coarse-

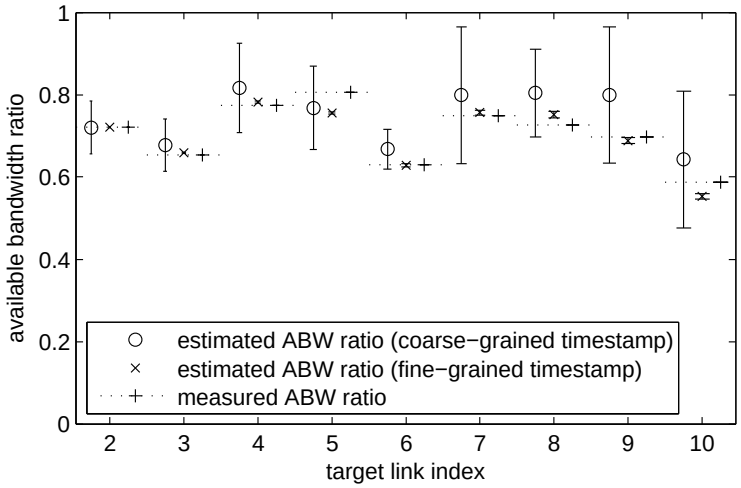


Figure 6: Accuracy of the proposed available bandwidth ratio estimation scheme for various target links under self-similar traffic loads

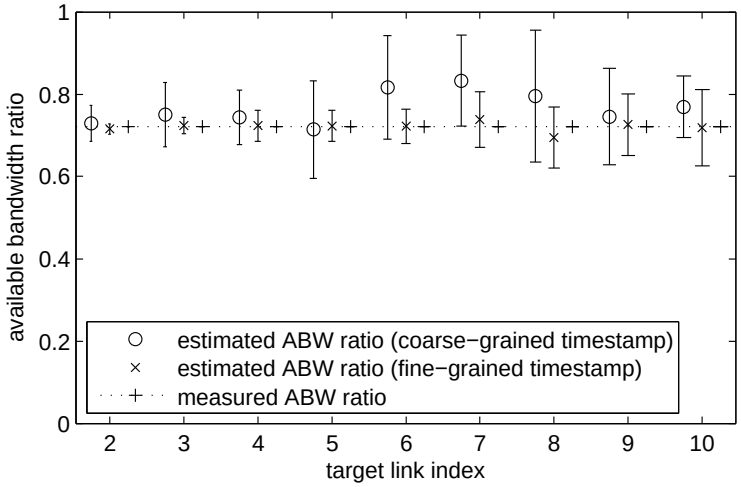


Figure 7: Accuracy of the proposed available bandwidth ratio estimation scheme for various target links under TCP traffic loads

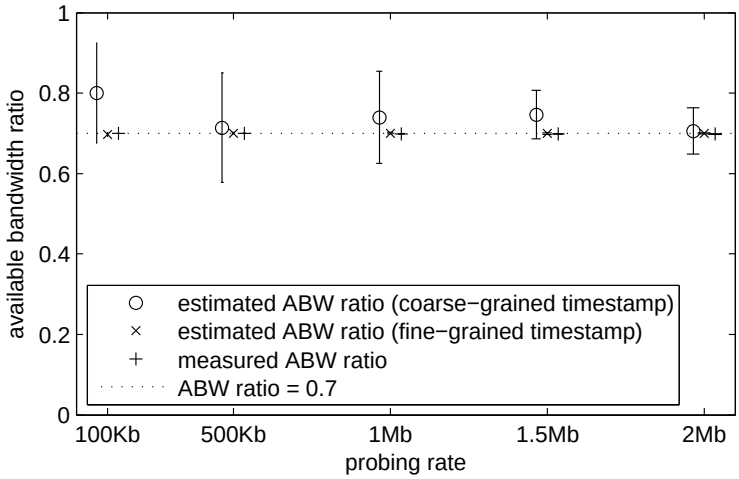


Figure 8: Accuracy of the proposed available bandwidth ratio estimation scheme for various probing rates under Poisson traffic loads (Target link: 2-nd link)

grained timestamps improves as the probing rate or the number of probe packets increases.

Thus far, we considered the cases where the load of the cross traffic on each link is less than 0.5 since the link utilizations are usually lower than 50% due to bandwidth over-provisioning in the core network [36]. We also investigate the performance of the proposed scheme for a possibly higher cross traffic load. Fig. 9 shows the accuracy of the proposed scheme when Poisson traffic is offered to each link at the load of 0.55. We find that a reasonable accuracy might be obtained for nearby links, e.g. links 2, 3, and 4. However, the estimation error of the scheme based on coarse-grained timestamps increases as the hop distance between the monitoring node and the target link increases, and we find the estimation results do not become reliable estimates of the available bandwidth ratios for the distant links, e.g. links 7 through 10. Thus, the number of hops that can be reliably probed by the scheme based on coarse-grained timestamps decreases as the traffic load on each link increases, and this issue has been mentioned in the discussion around (35) and (36) already.

The increase of the estimation error, especially for the scheme based on coarse-grained timestamps, can be explained as follows. As the traffic load on each link increases and the number of hops traversed by probe packets increases, the number of probe packets experiencing minimal end-to-end delay, i.e. within Δ from the minimum delay $D'_{m(0,i)}$, on the given path decreases by (33). Then, the value of j in (30) and (31) increases accordingly. Since $j\Delta$ corresponds to x in (28), the increase of j means the increase of x. Thus, we can easily expect that the estimation error will increase as the traffic load on each link increases, or the number of hops traversed by the probe packets increases from (28).

Fig. 9 also shows the upper bound of the available bandwidth ratio given by (28). We first obtained the maximum value of j defined in (31) for each link after 20 iterations of ns-2 simulation: the detailed values of the largest j's were measured to be $\{2, 2, 3, 5, 6, 7, 9, 10, 11\}$ for the 2-nd, 3-rd, 4-th, 5-th, 6-th, 7-th, 8-th, 9-th, and 10-th link, respectively. We next evaluated the upper bound on the right hand side of (28) for $x = j\Delta$, where Δ is 1 usec. We find that the upper bound of (28) tracks the increasing trend of the estimation error as the hop distance between the

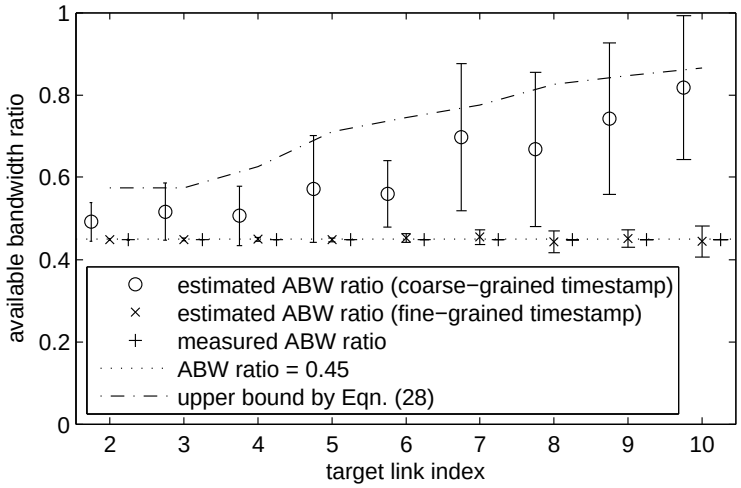


Figure 9: Accuracy of the proposed available bandwidth ratio estimation scheme for various links under a high load of Poisson traffic (Load of Poisson traffic = 0.55)

monitoring node and the target node increases, although it is not a strict upper bound. The upper bound of (28) is obtained from the assumption that the queueing delay at the target node has an exponential tail as described in (27), which is valid for an $M/M/1$ queueing system. Since this assumption may not be valid in general, the upper bound of (28) may not be a strict upper bound.

The estimation error for distant links, especially under a high traffic load, might be lowered by increasing the number of probe packets sent in one probing period as shown in a similar case of Fig. 8. However, the number of probing packets needs to be limited to maintain the probing traffic overhead low. Thus, the number of hops that can be probed by the coarse-grained timestamp-based scheme is dependent on the load of each link. Further extension of the probe coverage under the limited number of probe packets is left as a future research topic.

Thus far, the values of Δ, the resolution of the packet sending time, and Ω, the resolution of the timestamps provided by the remote nodes, were fixed to 1 usec and 1 msec, respectively. We now investigate how the accuracy of the coarse-grained timestamp-based scheme is affected by the resolution of the timestamps provided by the remote nodes, Ω. Fig. 10 shows the accuracy of the proposed scheme for three different values of Ω under Poisson traffic load. The environment is the same as that for Fig. 5, and we consider the available bandwidth ratio of the 9-th link. The accuracy of the fine-grained timestamp-based scheme is not affected by Ω, since the scheme does not depend on Ω. For the coarse-grained timestamp-based scheme, the estimation gets more accurate with a lower variance as the value of Ω decreases, and the estimation values tend to overestimate the available bandwidth ratio as Ω increases. If Ω increases for a fixed value of Δ, the number of probe packets experiencing the minimal delay is likely to decrease by (33). Then, the value of j in (30) and (31) will increase accordingly. Since the increase of j means the increase of x in (25), the coarse-grained timestamp-based scheme tends to overestimate the available bandwidth ratio in this case by the reason given around (25) and (26). On the contrary, the smaller value of Ω leads to the increase of number of probe packets experiencing the minimal delay among the given number of probe packets. Thus, the value of j in (30) and (31) can be maintained low as an integer close to 1, and a more accurate result can be obtained with a larger

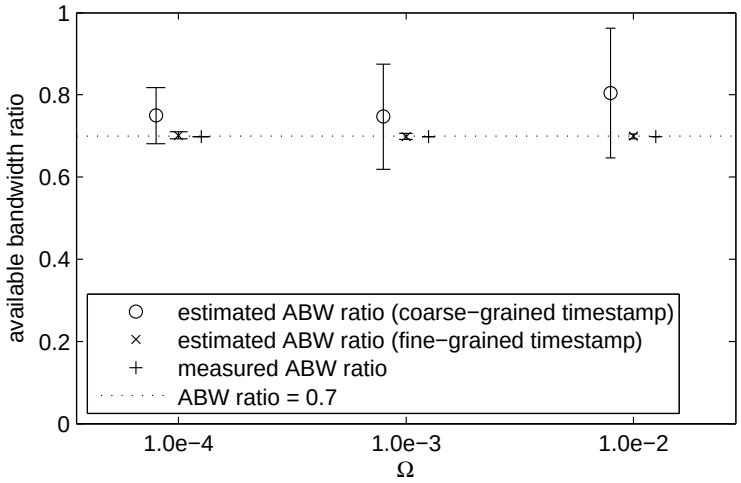


Figure 10: Effect of Ω, i.e. the resolution of the timestamps provided by the remote nodes, on the estimation accuracy under Poisson traffic loads (Target link: 9-th link)

value in the numerator and the denominator of (30) as Ω decreases. We also find that the probing rate might be reduced accordingly as Ω decreases, i.e. more accurate timestamps are provided by the remote nodes, by the reason explained above.

We evaluate the performance of the proposed available bandwidth ratio estimation scheme in a more dynamic environment. The target link is the 7-th link between CR6 and CR7 in Fig. 4. 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 self-similar traffic loads. We find that the estimation results are close to the measured available bandwidth ratio when fine grained timestamps are returned by the remote nodes. Although the estimation accuracy degrades under the coarse-grained timestamp environment, the estimated values follow the changes of the available bandwidth ratio most of the time. Fig. 12 shows the performance of the proposed scheme under time-varying TCP (Reno) traffic. In the simulation, the TCP flows arrive at the 7-th link according to Poisson process, and their duration follows the Pareto distribution with a shape parameter of 1.5. We set the average duration of the flow to 35.56 seconds. The arrival rate is chosen to generate a traffic load of about 0.5 on average. It is known that real TCP flows arrive according to Poisson process and their durations have a heavy-tailed distribution [56]. We also observe that estimated values track the measured available bandwidth ratios.

We next 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 are retained 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 for the case shown in Fig. 11. The target link is the 7-th link and a self-similar traffic load is offered to that link. Fig. 13 shows the test result for this case. The tendency is very similar to the case of Fig. 11. The proposed scheme closely tracks the change of the available bandwidth although the target link is behind the tight link. Fig. 14 shows the results

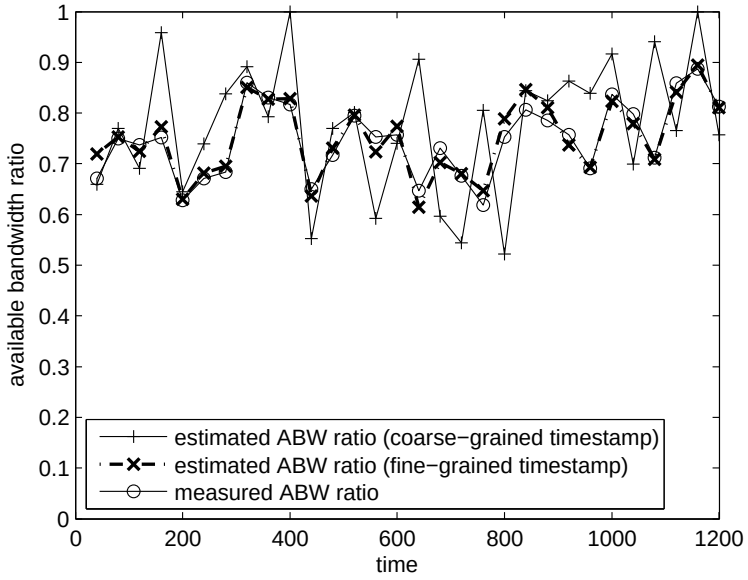


Figure 11: Comparison of the measured and estimated available bandwidth ratios under self-similar traffic loads

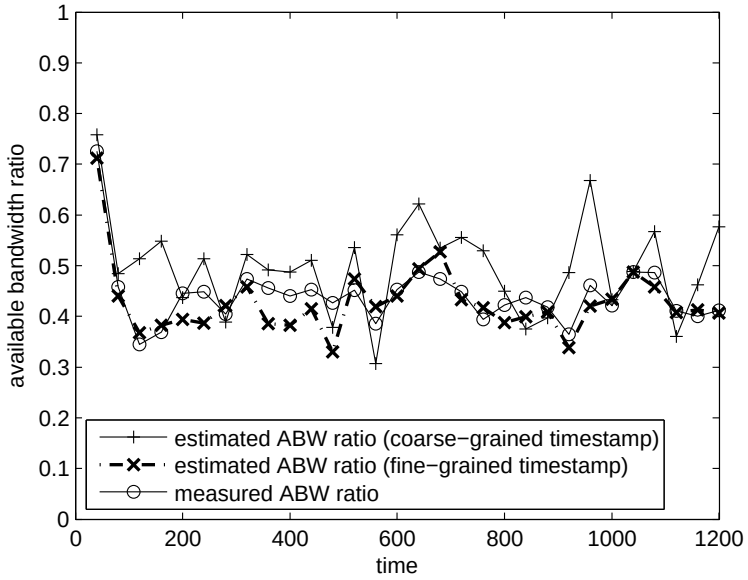


Figure 12: Comparison of the measured and estimated available bandwidth ratios under TCP traffic loads

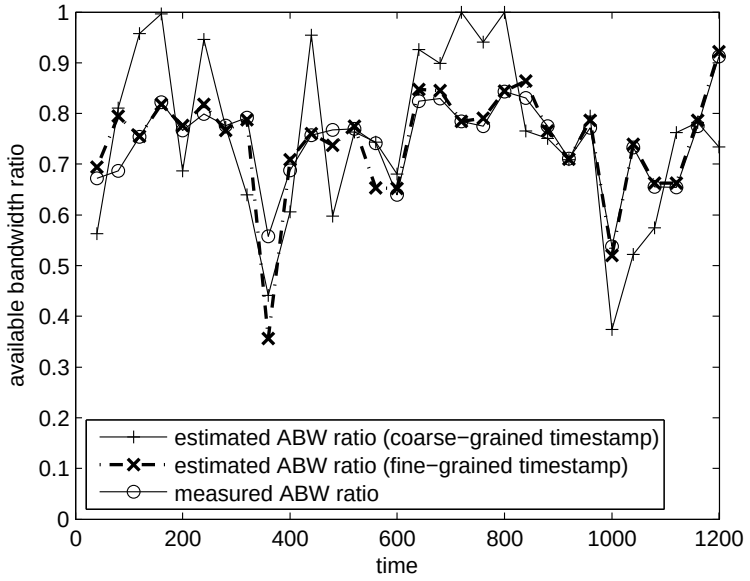


Figure 13: Available bandwidth ratio estimation for a link (7-th link) behind the tight link (5-th link) under self-similar traffic loads

under TCP traffic loads. The estimated results are close to the measured available bandwidth ratio when fine-grained timestamps are available. Although the accuracy degrades compared with Fig. 12 under the coarse-grained timestamp environment, the estimated values still track the change of available bandwidth ratio. The accuracy degradation for the link behind the tight link under coarse-grained timestamps can be explained as follows. If the link rate of 5-th link is 150 Mbps, each probe packet is likely to stay longer on the tight link, i.e. 5-th link, compared with the case when the link rate of 5-th link is 1 Gbps because of the reduced available bandwidth on that link. Then, j in (31) might increase due to the increased delay of probe packets. The increase of j in (26) or (30), or equivalently x in (25), usually leads to overestimation of the available bandwidth ratio by the reason given around (25) and (26).

Thus far, we consider the cases where the cross traffic interferes with the probing stream only at one hop. We finally evaluate the proposed available bandwidth estimation scheme for the cross traffic streams interfering with the probing stream at multiple links. The detailed scenario is as follows. All the link rates are fixed to 1 Gbps again. Node Si ($i \leq 8$) sends three self-similar traffic streams with the same average rate of r: the first stream to Node Di, the second stream to Node D{i+1}, and the third stream to Node D{i+2}, respectively. Node S9 sends only two streams: one stream to D9 and another stream to D10. Node S10 sends only one stream to D10. Then, the average cross traffic rate becomes about $3r$ on the first and the 10-th links, $5r$ on the second and the 9-th links, and $6r$ on all the other middle links. We fix the value of r to 60 Mbps, and thus, the load on the middle links is around $6r = 360$ Mbps. Fig. 15 shows the simulation results obtained under this environment. We find that the measured available bandwidth ratio is within the standard deviation from the average value of the estimated available bandwidth ratio on most links for both fine-grained timestamp-based and coarse-grained timestamp-based schemes.

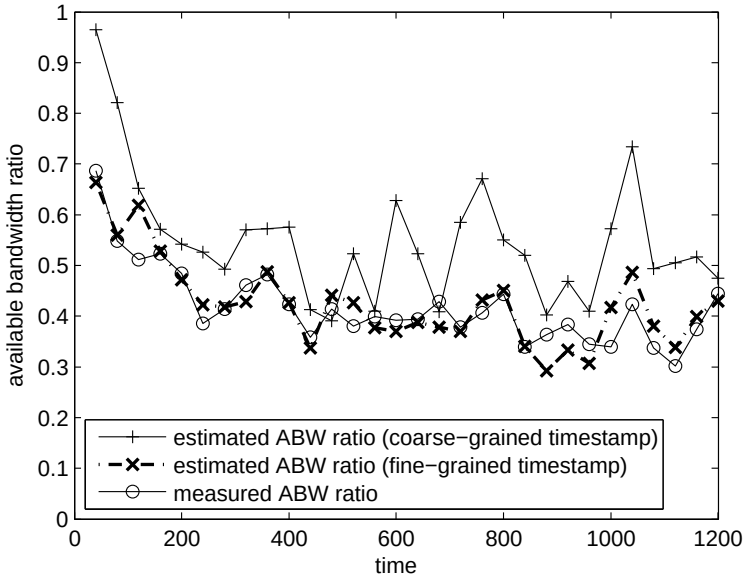


Figure 14: Available bandwidth ratio estimation for a link (7-th link) behind the tight link (5-th link) under TCP traffic loads

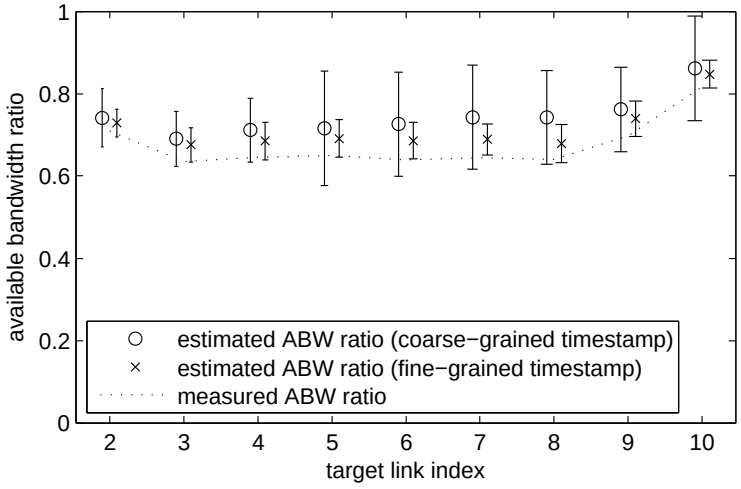


Figure 15: Accuracy of the proposed available bandwidth ratio estimation scheme for various target links under self-similar traffic streams interfering with the probing traffic stream at multiple links

6. Conclusion

In this paper, we proposed a scheme which estimates the available bandwidth ratio of a remote link or path segments without deploying any special tool at the remote nodes. We measure the one-way delay from the difference of the packet sending time and the timestamp value received from the remote nodes, extract the queueing delay component from the measured delay, and 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. We use ICMP timestamp packets to measure one-way delays, but the use of ICMP timestamps entails some major challenges. One of them is the coarse resolution (1 msec) of timestamps and another is the clock skew between different nodes. We first develop a technique which can estimate the available bandwidth ratio of a remote link under accurate timestamps provided by the remote nodes. We next investigate a statistical method to extract the queueing delay distribution from the coarse resolution delays and then develop a mechanism to estimate the available bandwidth ratio of a remote link using the coarse-grained timestamps provided by the remote nodes attached to the target link. The clock skew problem is addressed with existing approaches. There is one more obstacle that needs to be resolved before applying the proposed approach in the production network. The ICMP timestamp does not reflect the real packet arrival time at the target node because the ICMP packet response is treated on the slow path in router packet processing and the processing time is not fixed. This problem might be resolved by timestamping in the NIC or incorporating ICMP timestamping into the fast path in router packet processing. The detailed implementation issue will be discussed further in future study.

Since our scheme needs not incur a short-term congestion unlike conventional end-to-end available bandwidth estimation mechanisms, the intrusiveness is low and the proposed scheme overcomes the limitation of conventional approaches, inability to probe the links beyond the narrow or the tight link. 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.

Appendix A. Derivation of (19)

When $j = 1$ in (18), we can obtain

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

The piecewise linear assumption for $\Pr(Q_{0,i} \le t)$ in Section 4 can be restated as

$$\Pr(Q_{0,i} \le t) \approx \alpha_k t + \beta_k, \qquad t \in [k\Delta, (k+1)\Delta]. \tag{A.2}$$

Combining (A.1) and (A.2) yields

$$\begin{aligned} \Pr(D'_{0,i} - D'_{m(0,i)} \le \Delta) &\approx \frac{1}{\Omega} \int_0^{\Delta} (\alpha_0 t + \beta_0)\, dt \\ &= \frac{\Delta}{\Omega} \left(\frac{1}{2} \alpha_0 \Delta + \beta \right). \end{aligned} \tag{A.3}$$

Since $\Pr(Q_{0,i} \le \Delta/2) \approx \alpha_0\Delta/2 + \beta_0$ by (A.2), (A.3) can be changed into

$$\Pr(Q_{0,i} \le \Delta/2) \approx \frac{\Omega}{\Delta} \Pr(D'_{0,i} - D'_{m(0,i)} \le \Delta), \tag{A.4}$$

which is the first relation of (19).

When $j = n-1$ and $j = n$ in (18), we can obtain

$$\Pr(D'_{0,i} - D'_{m(0,i)} \le (n-1)\Delta) = \frac{1}{\Omega} \sum_{k=0}^{n-2} \int_{k\Delta}^{(k+1)\Delta} \Pr(Q_{0,i} \le t)dt, \quad 2 \le n < \Omega/\Delta + 1,$$

$$\Pr(D'_{0,i} - D'_{m(0,i)} \le n\Delta) = \frac{1}{\Omega} \sum_{k=0}^{n-1} \int_{k\Delta}^{(k+1)\Delta} \Pr(Q_{0,i} \le t)dt, \quad 1 \le n < \Omega/\Delta.$$

By subtracting the upper equation from the lower one in the above equation set, we obtain

$$\Pr(D'_{0,i} - D'_{m(0,i)} \le n\Delta) - \Pr(D'_{0,i} - D'_{m(0,i)} \le (n-1)\Delta) = \frac{1}{\Omega} \int_{(n-1)\Delta}^{n\Delta} \Pr(Q_{0,i} \le t)dt, \quad 2 \le n < \Omega/\Delta. \tag{A.5}$$

In a similar way to derivation of (A.3) and (A.4), the following relation can be derived using the piecewise linear assumption for $\Pr(Q_{0,i} \le t)$:

$$\int_{(n-1)\Delta}^{n\Delta} \Pr(Q_{0,i} \le t)dt \approx \Delta \cdot \Pr\left(Q_{0,i} \le (n-1/2)\Delta\right). \tag{A.6}$$

Combining (A.5) and (A.6) yields

$$\Pr\left(Q_{0,i} \le (n-1/2)\Delta\right) \approx \frac{\Omega}{\Delta} \left\{\Pr(D'_{0,i} - D'_{m(0,i)} \le n\Delta) - \Pr(D'_{0,i} - D'_{m(0,i)} \le (n-1)\Delta)\right\}, \quad 2 \le n < \Omega/\Delta, \tag{A.7}$$

which is the second relation of (19).

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, Z. Su, S. F. Wu, and F. T. Chong, "On deriving unknown vulnerabilities from zeroday polymorphic and metamorphic worm exploits," *in Proc. ACM CCS*, Alexandria, Virginia, Nov. 2005.

[3] R. Mahajan, N. Spring, D. Wetherall, and T. Anderson, "User-level internet path diagnosis," *in Proc. Symposium on Operating Systems Principles (SOSP)*, Oct. 2003.

[4] K. G. Anagnostakis, M. Greenwald, and R. S. Ryger, "cing: Measuring network-internal delays using only existing infrastructure," *in Proc. IEEE INFOCOM*, pp. 2112-2121, Apr. 2003.

[5] Y. Tang, E. Al-Shaer, R. Boutaba, "Efficient fault diagnosis using incremental alarm correlation and active investigation for Internet and overlay networks," *IEEE Transactions on Network and Service Management*, vol. 5, no. 1, pp. 36-49, Mar. 2008.

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

[7] A. Shriram, M. Murray, Y. Hyun, N. Brownlee, A. Broido, M. Fomenkov, and kc claffy, "Comparison of public end-to-end bandwidth estimation tools on high-speed links," *in Proc. PAM*, Mar.-Apr. 2005.

[8] R. L. Carter and M. E. Crovella, "Measuring bottleneck link speed in packet-switched networks," *Performance Evaluation*, vol. 27-28, pp. 297-318, Oct. 1996.

[9] 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.

[10] M. Jain and C. Dovrolis, "Pathload: A measurement tool for end-to-end available bandwidth," *in Proc. PAM*, Mar. 2002.
[11] V. Ribeiro, R. H. Riedi, R. G. Baraniuk, J. Navratil, L. Cottrell, "pathChirp: Efficient available bandwidth estimation for network path," *in Proc. PAM*, Apr. 2003.
[12] N. Hu and P. Steenkiste, "Evaluation and Characterization of Available Bandwidth Probing Techniques," *IEEE JSAC*, vol. 21, no. 10, Aug. 2003.
[13] J. Strauss, D. Katabi, and F. Kaashoek, "A measurement study of available bandwidth estimation tools," *in Proc. ACM IMC*, Oct. 2003.
[14] S. Y. Nam, S. Kim, J. Kim, and D. K. Sung, "Probing-Based Estimation of End-to-End Available Bandwidth," *IEEE Communications Letters*, vol. 8, no. 10, June 2004.
[15] J. Sommers, P. Barford, and W. Willinger, "A proposed framework for calibration of available bandwidth estimation tools," *in Proc. of IEEE Symposium on Computers and Communications (ISCC)*, pp. 709-718, 2006.
[16] E. Goldoni, G. Rossi, and A. Torelli, "Assolo, a new method for available bandwidth estimation," *in Prof. of International Conference on Internet Monitoring (ICIMP)*, pp. 130-136, May 2009.
[17] E. Goldoni and M. Schivi, "End-to-end available bandwidth estimation tools, an experimental comparison," *in Prof. of Traffic Monitoring and Analysis (TMA) Workshop*, Zurich, Switzerland, Apr. 2010.
[18] S. Y. Nam, S. Kim, and W. Park, "Analysis of minimal backlogging-based available bandwidth estimation mechanism," *Computer Communications*, vol. 35, no. 4, pp. 431-443, Feb. 2012.
[19] K. Harfoush, A. Bestavros, and J. Byers, "Measuring bottleneck bandwidth of targeted path segments," *in Proc. of IEEE INFOCOM*, Mar.-Apr. 2003.
[20] R. Govindan and V. Paxson, "Estimating router ICMP generation delays," *in Proc. PAM*, Mar. 2002.
[21] V. Paxson, "On calibrating measurements of packet transit time," *in Proc. ACM SIGMETRICS*, pp. 11-21, June 1998.
[22] 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.
[23] L. Zhang, Z. Liu, and C. H. Xia, "Clock synchronization algorithms for network measurements," *in Proc. IEEE INFOCOM*, pp. 160-169, June 2002.
[24] R. S. Ryger, fixclock: Removing clock artifacts from communication timestamps, Technical Report DCS/TR-1243, Yale University, March 2003.
[25] T. Kohno, A. Broido, and kc claffy, "Remote physical device fingerprinting," *IEEE Tranactions on Dependable and Secure Computing*, vol. 2, no. 2, pp. 93-108, Apr. 2005.
[26] V. Jacobson, pathchar - a tool to infer characteristics of internet paths, available from ftp://ftp.ee.lbl.gov/pathchar, Apr. 1997.
[27] Y. He, M. Faloutsos, S. Krishnamurthy, and B. Huffaker, "On routing asymmetry in the internet," *in Proc. of IEEE Globecom*, Nov.-Dec. 2005.
[28] M.-F. Shih and A. Hero, "Unicast inference of network link delay distributions from edge measurements," *in Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing.*, Salt Lake City, UT, May 2001.
[29] M. J. Coates and R. D. Nowak, "Sequential Monte Carlo inference of internal delays in nonstationary data networks," *IEEE Trans. Signal Processing*, vol. 50, no 2, Feb. 2002.
[30] Y. Tsang, M. Coates, and R. D. Nowak, "Network delay tomography," *IEEE Trans. Signal Processing*, vol. 51, no 8, Aug. 2003.
[31] A. Rizk and M. Fidler, "On the identifiability of link service curves from end-host measurements," *in Proc. of NET-COOP*, Paris, France, Sep. 2008.
[32] F. Viger, Active Probing with ICMP Packets, Report, University of Melbourne, Jul. 2003.
[33] G. Lu, Y. Chen, S. Birrer, F.E. Bustamante, X. Li, "POPI: a user-level tool for inferring router packet forwarding priority," *IEEE/ACM Trans. Networking*, vol. 18, no 1, pp. 1-14, Feb. 2010.
[34] Endace, Endace Measurement Systems, http://dag.cs.waikato.ac.nz/, 2004.
[35] G. Jin, B. L. Tierney, "System capability effects on algorithms for network bandwidth measurement," *in Proc. ACM IMC*, Oct. 2003.
[36] C. Fraleigh, S. Moon, B. Lyles, C. Cotton, M. Khan, D. Moll, R. Rockell, T. Seely, and C. Diot, "Packet-level traffic measurements from the Sprint IP backbone," *IEEE Network*, vol. 17, no. 10, pp. 6-16, Nov.-Dec. 2003.
[37] M. Luckie and T. McGregor, "Path diagnosis with IPMP," *in Proc. ACM/SIGCOMM Network Troubleshooting Workshop*, Aug. 2004.
[38] Global Environment for Network Innovations (GENI), http://www.geni.net/.
[39] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, OpenFlow: Enabling Innovation in Campus Networks, Whitepaper, http://www.openflowswitch.org/, 2008.
[40] X. Liu, K. Ravindran, and D. Loguinov, "A stochastic foundation of available bandwidth estimation: multi-hop analysis," *IEEE/ACM Trans. Networking*, vol. 16, no. 1, Feb. 2008.
[41] X. Liu, K. Ravindran, B. Liu, and D. Loguinov, "Single-hop probing asymptotics in available bandwidth estimation:

sample-path analysis," *in Proc. of ACM IMC*, Oct. 2004.
[42] D. Antoniades, M. Athanatos, A. Papadogiannakis, E. P. Markatos, C. Dovrolis, "Available bandwidth measurement as simple as running wget," *in Proc. of PAM*, Mar. 2006.
[43] S. Chakravarty, A. Stavrou, A. D. Keromytis, LinkWidth: a method to measure link capacity and available bandwidth using single-end probes, Technical Report (CUCS-002-08), Department of Computer Science, Columbia University, Jan. 2008.
[44] N. Hu, L. Li, Z. M. Mao, P. Steenkiste, and J. Wang "Locating Internet bottlenecks: algorithms, measurements, and implications," *in Proc. of ACM SIGCOMM*, Mar. 2004.
[45] F. Baccelli, S. Machiraju, D. Veitch, and J. Bolot, "The role of PASTA in network measurement," *in Proc. of ACM SIGCOMM*, Sep. 2006.
[46] V. Paxson, "End-to-end routing behavior in the internet," *in Proc. of ACM SIGCOMM*, Aug. 1996.
[47] S. Chuang, A. Goel, N. McKeown, and B. Probhakar, "Matching output queueing with a combined input output queued switch," *in Proc. of IEEE INFOCOM*, Mar. 1999.
[48] H. Lee, and S. Seo, "Matching output queueing with a multiple input/output-queued switch," *in Proc. of IEEE INFOCOM*, Mar. 2004.
[49] R. W. Wolff, Stochastic modeling and the theory of queues, Prentice Hall, 1989.
[50] J. Walrand and P. Varaiya, "Sojourn times and the overtaking condition in Jacksonian networks," *Adv. Appl. Prob.*, vol. 12, no. 4, pp. 1000-1018, 1980.
[51] K.C. Sevcik and I. Mitrani, "The distribution of queueing network states at input and output instants," *Journal of the ACM*, vol. 28, no. 2, pp. 358-371, 1981.
[52] S. Y. Nam, S. Lee, and H. S. Kim, Estimation of available bandwidth of a remote link or path segments, CMU-CyLab-06-012, CMU, Jul. 2006.
[53] S. M. Ross, Probability Models for Computer Science, Harcourt/Academic Press, 2002.
[54] M. E. Crovella and A. Bestavros, "Self-similarity in World Wide Web traffic: evidence and possible causes," *IEEE/ACM Trans. Networking*, vol. 5, no. 10, Dec. 1997.
[55] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the self-similar nature of Ethernet traffic," *IEEE/ACM Trans. Networking*, vol. 2, no. 1, Feb. 1994.
[56] V. Paxson and S. Floyd, "Wide-area traffic: the failure of Poisson modeling," *IEEE/ACM Trans. Networking*, vol. 3, no. 3, June 1995.
[57] 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*, vol. 45, no. 3, Apr. 1999.