Quick End-to-End Available Bandwidth Estimation for QoS of Real-Time Multimedia Communication

Takashi Oshiba and Kazuaki Nakajima
Service Platforms Research Laboratories, NEC Corporation, Japan
{oshiba@cp, nakajima@ah}.jp.nec.com

***Abstract*—We propose PathQuick, a quick end-to-end available bandwidth estimation method. When PathQuick is used in real-time multimedia communication, such as real-time photo sharing and video conferencing, *just before* the transmission of media data such as photo and video, it can quickly complete the estimation of the latest available bandwidth. Consequently, the bandwidth consumption of media data can be controlled to within the available bandwidth estimated at the beginning of media data transmission, and this will prevent delay and packet loss. Thus, QoS at the beginning of real-time photo sharing and video conferencing can be ensured. Conventional methods have a critical restriction that they require a long estimation time; e.g., several seconds or several ten seconds. Using these methods *just before* the media data transmission causes an unacceptably long waiting time until media data transmission begins, thus degrading real-time responsiveness. PathQuick achieves quick estimation by using a probing packet train (i.e., a set of multiple probing packets) with distinctive features: each packet within the packet train is placed at an equal time interval, and each packet size linearly increases as the packet sequence proceeds. Our evaluation of PathQuick has shown that its estimation duration is several hundred milliseconds, which is more than four times as fast as a conventional method. We also found that its probable range is more than a hundred times as wide as that of the conventional method when a limited waiting time is set to avoid degrading real-time responsiveness.**

Keywords: available bandwidth, quick estimation, QoS, real-time multimedia communication

I. INTRODUCTION

Real-time multimedia communication over IP networks such as real-time photo sharing [1], video conferencing [2], e-learning [3] and online gaming [4] has gained in popularity in recent years. If the bandwidth consumption of media data (such as photo, video and so forth) exceeds the available bandwidth – i.e., physical capacity minus bandwidth being used during a certain time period [5] – the media data will suffer delay and packet loss. Consequently, the measurement of available bandwidth is of great importance for ensuring the QoS of real-time multimedia communication.

In this paper, we propose a method, PathQuick, to quickly estimate the end-to-end available bandwidth, which always fluctuates. When used in real-time multimedia communication *just before* media data transmission, PathQuick can quickly complete the estimation of the currently available bandwidth. As a result, based on the available bandwidth a real-time multimedia communication system can control the means of media data transmission at the beginning of the transmission. For example, when a sender end-host begins to transmit photo data in a real-time photo sharing system, the system can dynamically transcode the photo data in order to shrink its data size [6]. The system can perform the transcoding based on the currently available bandwidth and a user-defined permissible transfer time limit. This enables the sender to complete the transfer of the photo to a receiver within the permissible transfer time limit. Another example is a video conferencing system. When a sender begins to transmit video data, the system can determine the initial video bit-rate [7] based on the currently available bandwidth. Thus, PathQuick can help such systems to ensure the QoS without a long waiting time.

Conventional end-to-end available bandwidth estimation methods [5] have a critical restriction in that they require a long estimation time; e.g., several seconds or several ten seconds (see Section II for details). Using these methods *just before* media data transmission would cause an unacceptably long waiting time before media data transmission could begin, thus degrading real-time responsiveness. Therefore, they cannot be used *just before* media data transmission. In contrast, PathQuick enables quick estimation by using a probing packet train (i.e., a set of multiple probing packets) with distinctive features: each packet within the packet train is placed at an equal time interval, and each packet size linearly increases as the packet sequence goes forward.

II. RELATED WORK

Since RTCP [8] can only be used *during* media data transmission, it cannot be used *just before* media data transmission. Furthermore, since it can only handle UDP packets, it cannot handle TCP packets such as photo data.

Much prior work, which actively transmits probing packets, has been done on end-to-end available bandwidth estimation. These methods fall into two broad classes [5]: packet pair methods and packet train methods. Typical examples of the first class are Abing [9] and Spruce [10], and examples of the second class are pathChirp [11] and Pathload [12]. Since pathChirp is the most closely related to PathQuick, we compare these two methods in Section V.

These conventional packet pair/train methods are mainly designed for non-real-time applications [10][12], such as optimal route selection in overlay networks [13], server selection in content delivery networks (CDNs) [14], service level agreement verification [15] and so on. Long estimation duration (e.g., several seconds) is not a considerable problem for such non-real-time applications. However, it is a critical problem for real-time multimedia communication. It has been reported that the estimation durations of Abing, Spruce, pathChirp and Pathload are as much as 1.3 s, 11.0 s, 5.5 s and 7.0 to 22.0 s, respectively [16].

A. Packet Pair Methods

Although Abing takes a relatively short time (1.3 s), packet pair methods are less accurate than packet train methods [17], as empirical evaluations have confirmed [16]. Therefore, packet pair methods such as Abing and Spruce are not suitable for real-time multimedia communication.

B. Packet Train Methods

Both pathChirp and Pathload are based on the probe rate model (PRM) [18]. PRM is based on the observation that (a) if the probing rate of a packet train at a sender is less than the available bandwidth, the probing packets will face no queuing delay at routers, so the time interval for each probing packet observed at a receiver will be the same as at the sender. On the other hand, (b) if the probing rate exceeds the available bandwidth, the packets will be queued at some router, increasing the time intervals observed at the receiver. The available bandwidth can be estimated by observing the probing rate at which there is a transition from (a) to (b).

1) Problem with pathChirp

One cause of pathChirp's long estimation latency (5.5 s) is the packet train structure; each time interval exponentially decreases as the packet sequence goes forward, and all packet sizes are equal. Thus, the per-packet probing rate exponentially increases within a single packet train. To probe over a wide range of rates, the time interval between the first and second packets must be long because this initial time interval determines the minimum probable bandwidth. The long initial time interval dominates the whole length of the packet train, resulting in a long estimation duration.

2) Problem with Pathload

Unlike pathChirp, Pathload employs constant bit-rate (CBR) packet trains; consequently, a single packet train may not determine the PRM transition point. Thus, it changes its probing rate, and repeatedly transmits packet trains with an iterative binary search algorithm to find the transition point, and this causes greater estimation latency (7.0 to 22.0 s). Although Pathload is similar to PathQuick in that both place each packet at an equal time interval, each packet size is equal in Pathload while each packet size varies in PathQuick.

3) Other Packet Train Methods

Several other packet train methods [19][20][21] for quick estimation have been recently proposed. However, the shortest estimation durations reported for these methods are 5.6 s [19], 10.0 s [20] and 20.0 s [21], respectively, so they are not suitable for real-time multimedia communication.

C. Variable Packet Size Probing

Pathchar [22], clink [23] and [24] are similar to PathQuick in that they vary the probing packet size. However, they estimate the physical capacity, which is fundamentally different from the available bandwidth [5]. Moreover, since pathchar and clink repeatedly transmit probing packets to complete the estimation, the estimation durations of pathchar and clink are long: 21 s and 300 s, respectively [25]. [26] also employs pathchar-based variable packet size probing, and its estimation duration is 225 s.

III. Requirements for Quick Estimation

Given the problems with conventional methods, particularly those described in Section II-B-1) and II-B-2), we identified two requirements that must be satisfied to enable quick available bandwidth estimation.

(1) **Short packet train length:** The whole transmission duration of a packet train must be short. PathChirp cannot satisfy this.

(2) **Wide coverage of probing range with a single packet train:** A single packet train must be able to probe over a wide range of rates to avoid repeated packet train transmissions. Pathload cannot satisfy this.

IV. Proposal of PathQuick

We propose PathQuick, a packet train method that satisfies both of the above requirements.

A. Design of Packet Train Structure

We designed the packet train structure of PathQuick with the following distinctive features.

(1) **Packet placement at equal time interval:** To satisfy the first requirement, the time interval for each packet within the packet train must be short. To this end, we designed the packet train so that each packet is placed at an equal time interval (see Fig. 1-(1)).

(2) **Linear increase of packet size:** To satisfy the second requirement, the per-packet probing rate must be changed within the single packet train. To this end, we designed the structure so that each packet size linearly increases from the previous one as the packet sequence proceeds (see Fig. 1-(2)). Note that the packet sizes in Pathload (and in pathChirp) are all the same.

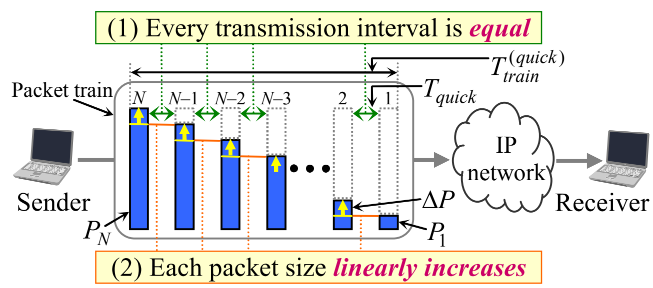

Figure 1. Design of packet train structure.

B. Mechanism of Quick Estimation

In PathQuick, a sender transmits a packet train of UDP probing packets to a receiver. Each packet carries a sender timestamp which the receiver uses along with its own local timestamp to estimate the available bandwidth. The receiver then estimates the available bandwidth and reports the estimated result to the sender by transmitting a UDP packet.

1) Packet Placement of Equal Time Interval

Let us consider a packet train consisting of N probing packets. Each packet is placed at equal time interval T_{quick} at the sender (Fig. 1-(1)). The whole transmission duration of a packet train (i.e., the packet train length) is

$$T_{train}^{(quick)} = T_{quick} \cdot (N-1) = T_{quick} \cdot N - T_{quick} \,. \quad (1)$$

For simplicity, we omit the transmission (or serialization) delay of each probing packet from $T_{train}^{(quick)}$. This omission does not matter since we assume T_{quick} is of the millisecond order, and the transmission delay is usually under the sub-millisecond order; i.e., short enough to be ignored.

Thus, packet train length $T_{train}^{(quick)}$ is a linear function of the number of probing packets N. As we will discuss in Section V-B-3), this $O(N)$ nature enables PathQuick to keep the packet train length shorter than that of pathChirp; thus, PathQuick satisfies the first requirement.

2) Linear Increase of Packet Size

The packet size of each probing packet is

$$P_i = P_1 + (i-1)\cdot \Delta P = \Delta P \cdot i + (P_1 - \Delta P), \quad (2)$$

where $i = 1, 2, \ldots, N$ and the constant value ΔP is the increase amount of the packet size (Fig. 1-(2)). Thus, each packet size P_i is a linear function of i, since P_1 is a constant value.

The per-packet probing rate at the i-th packet – i.e., the instantaneous probing rate of the packet train – is

$$R_i = \frac{P_i}{T_{quick}} = \frac{\Delta P}{T_{quick}} i + \frac{P_1 - \Delta P}{T_{quick}}. \quad (3)$$

Thus, each per-packet probing rate R_i is also a linear function of i. Therefore, PathQuick can increase the per-packet probing rate within a single packet train, and thereby can probe over a wide range of rates using a single packet train. Hence, PathQuick satisfies the second requirement.

C. PRM-based Available Bandwidth Estimation

Let us define the time interval between the i-th and $(i-1)$-th packet observed at the receiver as T_i^{rcv}, and the sender transmission time of the i-th packet as t_i.

Assuming CBR cross-traffic, the receiver analyzes the observed time intervals based on the PRM principle (see Section II-B) to estimate the available bandwidth as follows:

$$\begin{aligned} &\text{(a) } T_i^{rcv} = T_{quick}, \quad \text{if } R_i \le B[t_1, t_N] \\ &\text{(b) } T_{i-1}^{rcv} < T_i^{rcv}, \quad \text{otherwise,} \end{aligned} \quad (4)$$

where $B[t_1, t_N]$ is the actual available bandwidth between times t_1 and t_N. In PathQuick, a per-packet probing rate $R_k = P_k / T_{quick}$, where the k-th packet is the packet at which the observed time intervals at receiver T_i^{rcv} begin increasing, becomes the estimated available bandwidth. Fig. 2 illustrates the meaning of Eq. (4), $T_2^{rcv} = T_3^{rcv} =, \ldots, = T_{k-2}^{rcv} = T_{k-1}^{rcv} = T_{quick}$ in (a) and $T_{quick} < T_k^{rcv} < T_{k+1}^{rcv} <, \ldots, < T_N^{rcv}$ in (b). That is, the k-th packet is the transition point of PRM, and the per-packet probing rate of the k-th packet becomes the estimated available bandwidth. The above assumption of CBR cross-traffic, however, does not always hold true in real networks. Intermittent and bursty cross-traffic causes the queuing delay to fluctuate, so the observed time intervals may not increase monotonically. Hence, simply using Eq. (4) would lead to erroneous estimation results. To avoid this, we employ a technique called excursion segmentation of pathChirp [11].

V. Evaluation

We evaluated PathQuick regarding various aspects. In the evaluation, we compared PathQuick with pathChirp since pathChirp is the method most closely related to PathQuick.

A. Parameter Choice for Quantitative Evaluation

Before going into details of the evaluation, we will explain how we choose the values of several parameters used for the quantitative evaluation.

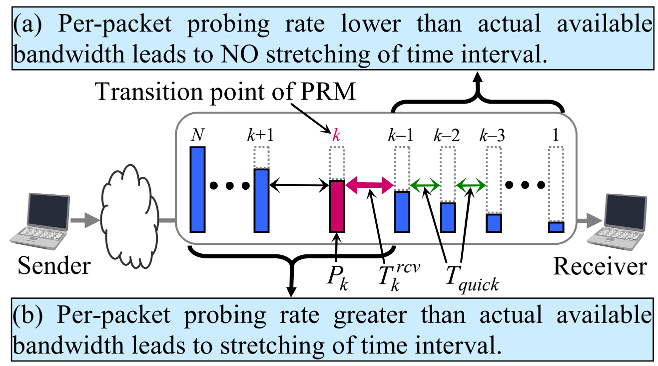

Figure 2. PRM-based available bandwidth estimation.

1) Maximum Probable Bandwidth

Akamai, a major global CDN provider of video and Web content, has reported in Fig. 17 of [27] that the percentage of throughput at less than 10 Mbps over the current Internet in the United States, Sweden and Japan (which are representative broadband countries in North America, Europe and Asia) is 95.4%, 87.6% and 81.4%, respectively. This implies that most of the end-to-end available bandwidth over the current Internet is less than 10 Mbps. Hence, we choose 10 Mbps as a minimum target for the maximum probable bandwidth in our evaluation.

2) RTT

Since the average one-way delay (OWD) in current Japanese Internet use among 13 major Japanese cities has been reported to be 26.21 ms in Table 1 of [28], we set the end-to-end round trip time (RTT) to 52 ms in our evaluation.

B. Estimation Duration

We analyzed the estimation duration of PathQuick and pathChirp under the same RTT and probable range. The estimation duration is the sum of the packet train length, queuing delay at routers, and RTT. Similar to the transmission delay for each probing packet in Section IV-B-1), we find in the same simulations of Section V-D that the queuing delay at routers is short enough to omit it from the estimation duration. Since we set the same RTT, we focus on the comparison between the packet train length of PathQuick $T_{train}^{(quick)}$ and that of pathChirp $T_{train}^{(chirp)}$.

1) Packet Train Length of PathQuick

Let us define the minimum and maximum probable bandwidths as $B_{\min}$ and $B_{\max}$, respectively. First, we formulate N as a function of P_1, P_N, $B_{\min}$ and $B_{\max}$, and then we formulate $T_{train}^{(quick)}$ as a function of P_1, P_N, $B_{\min}$ and $B_{\max}$ as follows. Obviously,

$$T_{quick} = \frac{P_N}{B_{\max}}. \quad (5)$$

The size of second packet is

$$P_2 = P_1 + \Delta P = P_1 + \frac{P_N - P_1}{N-1} = \frac{(N-2)P_1 + P_N}{n-1}. \quad (6)$$

Eqs. (5) and (6) lead to

$$B_{\min} = \frac{P_2}{T_{quick}} = \frac{B_{\max}}{P_N} \cdot \frac{(N-2)P_1 + P_N}{n-1}. \quad (7)$$

By solving Eq. (7) by N,

$$N = \frac{P_N(B_{\min} + B_{\max}) - 2P_1 B_{\max}}{P_N B_{\min} - P_1 B_{\max}}. \quad (8)$$

Hence, with Eqs. (1), (5) and (8), we have

$$T_{train}^{(quick)} = T_{quick} \cdot (N-1) = \frac{P_N(P_N - P_1)}{P_N B_{\min} - P_1 B_{\max}} . \quad (9)$$

2) Packet Train Length of pathChirp

Recall that pathChirp employs a probing packet train where each packet in the packet train is exponentially spaced, and all packet sizes are equal. Fig. 3 shows the differences in the packet train structures of PathQuick and pathChirp. The first time interval of the probing packets is T_{chirp}. The spread factor γ controls the exponential spacing. The packet size is P_{chirp}. Similar to PathQuick, first we formulate the number of probing packets in a packet train M as a function of γ, $B_{\min}$ and $B_{\max}$, and then we formulate $T_{train}^{(chirp)}$ as a function of P_{chirp}, γ, $B_{\min}$ and $B_{\max}$. Obviously,

$$T_{chirp} = \frac{P_{chirp}}{B_{\min}} . \quad (10)$$

Using Eq. (10),

$$B_{\max} = \frac{P_{chirp}\gamma^{M-2}}{T_{chirp}} = \frac{P_{chirp}\gamma^{M-2}}{P_{chirp}/B_{\min}} = \gamma^{M-2} B_{\min} . \quad (11)$$

Thus, $\gamma^{M-2} = B_{\max}/B_{\min}$ leads to $(M-2)\log\gamma = \log(B_{\max}/B_{\min})$ $= \log B_{\max} - \log B_{\min}$, so

$$M = \frac{\log B_{\max} - \log B_{\min}}{\log \gamma} + 2 . \quad (12)$$

Hence, with Eqs. (10) and (12), we have

$$T_{train}^{(chirp)} = T_{chirp} \sum_{i=2}^{M} \frac{1}{\gamma^{i-2}} = T_{chirp} \frac{1-(1/\gamma)^{M-1}}{1-1/\gamma} = \frac{P_{chirp}}{B_{\min}} \cdot \frac{1-(1/\gamma)^{\frac{\log B_{\max} - \log B_{\min}}{\log\gamma}+1}}{1-1/\gamma} . \quad (13)$$

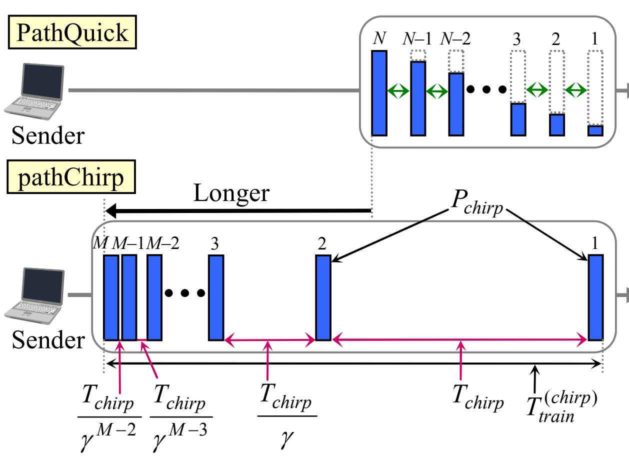

Figure 3. Packet train structure of PathQuick and pathChirp.

3) Comparison of Packet Train Length

We compared the packet train length of PathQuick $T_{train}^{(quick)}$ with that of pathChirp $T_{train}^{(chirp)}$. We set values to parameters such as T_{quick}, ΔP, P_1, $B_{\min}$, P_N, γ and P_{chirp} so that we could compare $T_{train}^{(quick)}$ with $T_{train}^{(chirp)}$ as a function of $B_{\max}$. That is, we compared the two packet train lengths under conditions where they would have the same probable range.

First, we set values to the parameters. To realize more than 10-Mbps target of probable range, we set $T_{quick} = 1$ ms, $P_1 = 1$ byte and $\Delta P = 12$ bytes. Then, according to Eq. (7), $B_{\min} = 8 \times (1+12)/0.001 = 104$ kbps. According to Eq. (5) and $T_{quick} = 1$ ms, now $B_{\max}$ depends on P_N. We set $P_N = 1{,}489$ bytes, which is the maximum value within the 1,500-bytes maximum transmission unit (MTU), to maximize $B_{\max}$. This means a packet train of PathQuick consists of $N = 1 + (1{,}489 - 1)/12 = 125$ packets. According to Eq. (5), $B_{\max} = 8 \times 1{,}489/0.001 = 11{,}912$ kbps. This successfully covers the 10-Mbps target. Consequently, in this comparison, $2 \le N \le 125$ and $104 \le B_{\max} \le 11{,}912$.

Since the developers of pathChirp argue the default value $\gamma = 1.2$ enables the highest accuracy [11], we used the default value. They also recommend $P_{chirp} \ge 1{,}000$ bytes [11], so we set $P_{chirp} = P_N = 1{,}489$ bytes.

Fig. 4 shows the comparison results for $T_{train}^{(quick)}(P_1, P_N, B_{\min}, B_{\max}) = T_{train}^{(quick)}(1, 1489, 104, B_{\max})$ and $T_{train}^{(chirp)}(\lambda, P_{chirp}, B_{\min}, B_{\max}) = T_{train}^{(chirp)}(1.2, 1489, 104, B_{\max})$, where $B_{\max}$ varies in the range $104 \le B_{\max} \le 11{,}912$. Due to the $O(N)$ nature of $T_{train}^{(quick)}$, PathQuick outperformed pathChirp. If $B_{\max} = 11{,}912$ kbps, then $T_{train}^{(quick)} = 124$ ms and $T_{train}^{(chirp)} = 682$ ms. Hence, the packet train length of pathChirp is 682 / 124 = 5.5 times longer than that of PathQuick. For 52-ms RTT, the estimation duration of PathQuick is 124+52=176 ms, and that of pathChirp is 682+52=734 ms, so PathQuick can complete the estimation 734 / 176 = 4.2 times as quickly as pathChirp.

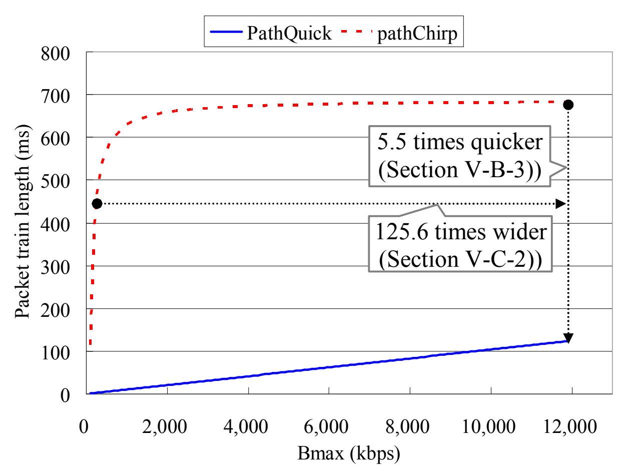

Figure 4. Comparison of packet train length.

C. Probable Range with a Single Packet Train

1) Estimation Time Limit

We compared the probable range with a single packet train for both methods under the condition that they had the same time limit. This time limit is important for real-time responsiveness in real-time multimedia communication. To explain this, let us consider a case of real-time photo sharing using voice-over-IP (VoIP). In VoIP, OWD, which means the inter-end-host synchronization error, must be less than 400 ms [29], so the inter-end-host synchronization error of

VoIP ranges from 0 to 400 ms. In addition, the inter-media synchronization error between audio and still images must be less than 500 ms [30]. If this is combined with the inter-end-host synchronization error, the limit of the inter-media synchronization error for the other party becomes a range from 500 ms to 900 ms. Here, assuming VoIP OWD is 0 ms leads to the most restricted condition: the limit of inter-media synchronization error for the other party becomes 500 ms.

In this context, the estimation duration of pathChirp, 734 ms as given in Section V-B-3), causes an unacceptable waiting time, resulting in an excessive synchronization error.

2) Comparison of Probable Range

Based on the consideration of synchronization error, we set the time limit of estimation duration to 500 ms. From Fig. 4, at 448 ms (500 ms minus the 52-ms RTT) PathQuick has completed estimation. Thus, the probable range of PathQuick is from 104 kbps to 11,912 kbps.

PathChirp, on the other hand, is still transmitting at 448 ms. Indeed, pathChirp transmits the third packet at 389 ms and the fourth one at 486 ms. Hence, under this time limit, pathChirp can use just three packets for estimation. The probable range of pathChirp is only from 104 kbps to 200 kbps. Hence, the probable range of PathQuick under this time limit is $(11{,}912-104)/(200-104)=125.6$ times as wide.

D. Estimation Accuracy

1) Experimental Setup

We did simulations to evaluate the estimation accuracy of the two methods with an ns-2 network simulator. Fig. 5 shows the topology, with a single bottleneck link (12 Mbps), of the simulations. The physical capacity and OWD of each link are also shown in Fig. 5. We set the RTT to 52 ms. We employed Poisson cross-traffic with a 1,000-byte packet size. The cross-traffic load was varied from 0 to 12 Mbps. We recorded the estimation results for each method. We used the same parameters described in Section V-A and V-B. In addition, we used default values of other pathChirp parameters, such as the busy period threshold and decrease factor described in [11].

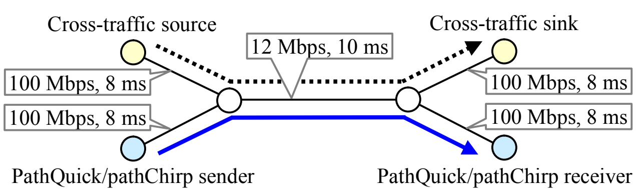

Figure 5. Topology with a single bottleneck link.

2) Estimation Results

Fig. 6 shows the estimation results for both methods. The sloping line shows the actual available bandwidth. We found that PathQuick outperformed pathChirp. In the range below 9 Mbps, PathQuick estimated the available bandwidth with an estimation error of approximately ± 2 Mbps. In contrast, the estimation result of pathChirp was highly erroneous, especially above 5 Mbps.

E. Resolution of Measurement

We analyzed the cause of accuracy difference between the both methods in Section V-D-2), and identified that the difference of resolution of measurement makes the accuracy difference.

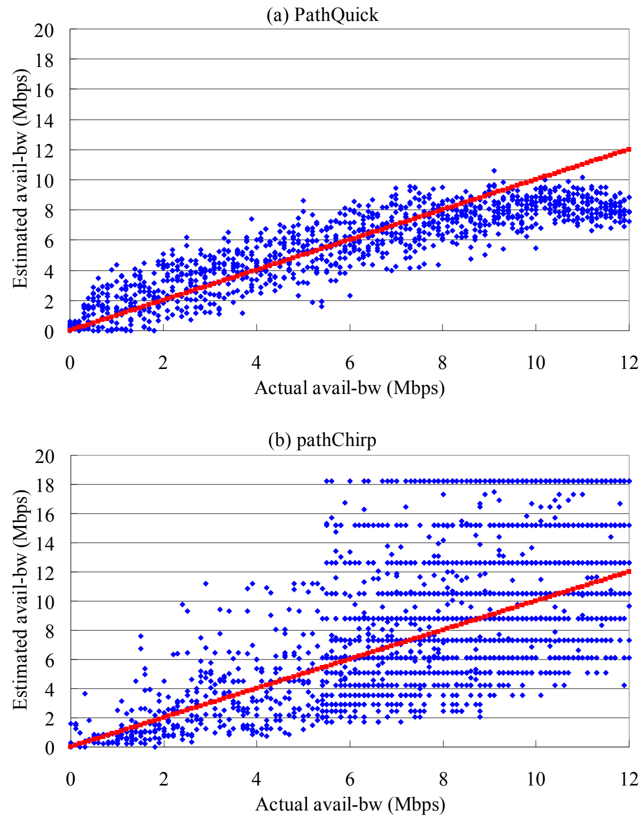

Figure 6. Estimation results for (a) PathQuick and (b) pathChirp.

We define a per-packet resolution of measurement as the gap of per-packet probing rate between neighbor packets. Note that the per-packet resolution of measurement directly affects the estimation accuracy but is not the estimation accuracy itself. According to Eq. (3), the per-packet resolution of PathQuick is

$$R_i - R_{i-1} = \frac{\Delta P}{T_{quick}} i + \frac{P_1 - \Delta P}{T_{quick}} - \frac{\Delta P}{T_{quick}}(i-1) - \frac{P_1 - \Delta P}{T_{quick}} = \frac{\Delta P}{T_{quick}} . \quad (14)$$

Thus, PathQuick has a constant resolution. On the other hand, according to Fig. 3, the per-packet resolution of pathChirp is

$$\frac{P_{chirp}\gamma^{i-2}}{T_{chirp}} - \frac{P_{chirp}\gamma^{i-3}}{T_{chirp}} = \frac{P_{chirp}(\gamma-1)}{T_{chirp}}\gamma^{i-3} = \frac{P_{chirp}(\gamma-1)}{T_{chirp}\gamma^3}\gamma^i . \quad (15)$$

Thus, pathChirp has an exponential (i.e., variable) per-packet resolution within a packet train. If the actual available bandwidth has high bit-rate, the transition point of PRM is detected by the tail part of a packet train. Since the per-packet resolution at the tail part of a packet train of pathChirp exponentially becomes coarse-grained, the highly erroneous estimation result was brought about in Fig. 6-(b).

We compared the average resolution of measurement of a single packet train. PathQuick's average resolution is $\Delta P / T_{quick} = 8 \times 12 / 0.001 = 96.0$ kbps. With Eq. (15), pathChirp's average resolution is

$$\frac{P_{chirp}(\gamma-1)}{T_{chirp}\gamma^3} \cdot \frac{1}{M} \sum_{i=1}^{M} \gamma^i . \quad (16)$$

With Eq. (12), we obtain $M = 28$ packets. So, with Eq. (16), pathChirp's average resolution is 416.7 kbps. Hence, the average resolution of PathQuick is 416.7 / 96.0 = 4.3 times as fine-grained as that of pathChirp.

F. Intrusiveness of Measurement

We compared the total packet size of a single packet train (i.e., intrusiveness) between both methods. PathQuick's total packet size of a single packet train is 1+2+, … , +1,489 = $125 \times (1+1{,}489)$ / 2 = 93.1 KB. PathChirp's total packet size is $28 \times 1{,}489$=41.7 KB. Thus, the intrusiveness of pathChirp is 93.1 / 41.7 = 2.2 times as low as that of PathQuick.

However, using double the increase amount $\Delta P = 24$ bytes and a half number of probing packets in a packet train $N = 63$ packets, while $P_1 = 1$ byte, $P_N = 1{,}489$ bytes, $T_{quick} = 1$ ms and $B_{\max} = 11{,}912$ kbps remain unchanged, PathQuick's total packet size of a single packet train becomes $63 \times (1+1{,}489)$ / 2 = 46.9 KB. With this slightly higher intrusiveness than pathChirp, PathQuick's average resolution of measurement of a single packet train becomes $\Delta P / T_{quick} = 8 \times 24 / 0.001 = 192.0$ kbps which is still more fine-grained than that of pathChirp. With these settings, we confirmed the estimation accuracy of PathQuick again outperforms pathChirp with an ns-2 simulation.

VI. Discussion

A tradeoff between estimation duration and accuracy has been identified [17]: using fewer packet trains or shorter packet trains reduces the estimation latency, but with a penalty in terms of accuracy. This tradeoff truly exists. For example, as mentioned in Section II, pathChirp can take 5.5 s but with better accuracy [16], while in this paper it takes 734 ms but with worse accuracy. This difference in accuracy mainly comes from the number of packet trains. In [16], pathChirp transmits multiple packet trains and then averages the per-train estimation results to obtain the final estimate, as described in [11]. In this paper, however, to reduce this method's estimation duration, only a single packet train was transmitted. Through a preliminary simulation, we observe this difference in accuracy for PathQuick, too.

As shown in Section V-B-3) and in Fig. 6, though, PathQuick completes estimation 4.2 times as quickly as pathChirp, and PathQuick achieves better estimation accuracy than pathChirp. Consequently, PathQuick better manages the tradeoff than pathChirp.

VII. Conclusion and Future Work

PathQuick is a quick end-to-end available bandwidth estimation method. Through an evaluation, we confirmed that it completes the estimation in 176 ms, or 4.2 times as quickly as pathChirp. We also confirmed that its probable range is 125.6 times as wide as that of pathChirp under a condition of limited estimation duration. The estimation accuracy of PathQuick outperformed pathChirp, and the average resolution of measurement of a single packet train of PathQuick is 4.3 times as fine-grained as that of pathChirp.

In future work, we will conduct large-scale experiments over the Internet and also over commercial cellular networks to validate PathQuick in real network environments.

References

[1] Y. Sumi, J. Ito and T. Nishida, "Photochat: communication support system based on sharing photos and notes," *ACM CHI extended abstracts*, pp. 3237–3242, 2008.

[2] L. D. Cicco et al., "Skype video responsiveness to bandwidth variations," *ACM NOSSDAV*, pp. 81–86, 2008.

[3] V. Roesler et al., "A new multimedia synchronous distance learning system: the IVA study case," *ACM SAC*, pp. 1765–1770, 2009.

[4] J. Jardine and D. Zappala, "A hybrid architecture for massively multiplayer online games," *ACM NetGames*, pp. 60–65, 2008.

[5] R. Prasad, C. Dovrolis, M. Murray and K. Claffy, "Bandwidth estimation: metrics, measurement techniques, and tools," *IEEE Network*, Vol. 17, Issue 6, pp. 27–35, 2003.

[6] S. Coulombe and S. Pigeon, "Quality-aware selection of quality factor and scaling parameters in JPEG image transcoding," *IEEE CIMSVP*, pp. 68–74, 2009.

[7] T. Stockhammer, H. Jenkac and G. Kuhn, "Streaming video over variable bit-rate wireless channels," *IEEE Transactions on Multimedia*, Vol. 6, Issue 2, pp. 268–277, 2004.

[8] H. Schulzrinne et al., "RTP: a transport protocol for real-time applications," IETF RFC 3550, July 2003.

[9] J. Navratil and L. Cottrell, "ABwE: a practical approach to available bandwidth estimation," *PAM Workshop*, 2003.

[10] J. Strauss, D. Katabi and F. Kaashoek, "A measurement study of available bandwidth estimation tools," *ACM IMC*, pp. 39–44, 2003.

[11] V. J. Ribeiro, R. H. Riedi, R. G. Baraniuk, J. Navratil and L. Cottrell, "pathChirp: efficient available bandwidth estimation for network paths," *PAM Workshop*, 2003.

[12] M. Jain and C. Dovrolis, "End-to-end available bandwidth: measurement methodology, dynamics, and relation with TCP throughput," *ACM SIGCOMM*, pp. 295–308, 2002.

[13] D. Andersen, H. Balakrishnan, F. Kaashoek and R. Morris, "Resilient overlay networks," *ACM SOSP*, pp. 131–145, 2001.

[14] K. M. Hanna, N. Natarajan and B. N. Levine, "Evaluation of a novel two-step server selection metric," *IEEE ICNP*, pp. 290–300, 2001.

[15] Y. Cheng and W. Zhuang, "Dynamic inter-SLA resource sharing in path-oriented differentiated services networks," *IEEE/ACM Transactions on Networking*, Vol. 14, Issue 3, pp. 657–670, 2006.

[16] A. Shriram et al., "Comparison of public end-to-end bandwidth estimation tools on high-speed links," *PAM Workshop*, 2005.

[17] M. Jain and C. Dovrolis, "Ten fallacies and pitfalls on end-to-end available bandwidth estimation," *ACM IMC*, pp. 272–277, 2004.

[18] L. Lao, C. Dovrolis and M. Y. Sanadidi, "The probe gap model can underestimate the available bandwidth of multihop paths," *ACM SIGCOMM Computer Communication Review*, Vol. 36, Issue 5, pp. 29–34, 2006.

[19] D. Croce et al., "Fast available bandwidth sampling for ADSL links: rethinking the estimation for larger-scale measurements," *PAM Conference*, pp. 67–76, 2009.

[20] Q. Liu and J. Hwang, "End-to-end available bandwidth estimation and time measurement adjustment for multimedia QoS," *IEEE ICME*, Vol. 3, pp. 373–376, 2003.

[21] S. S. Wang et al., "Fast end-to-end available bandwidth estimation for real-time multimedia networking," *IEEE MMSP*, pp. 415–418, 2006.

[22] V. Jacobson, "Pathchar—a tool to infer characteristics of Internet paths," April 1997, [online] ftp://ftp.ee.lbl.gov/pathchar/msri-talk.pdf

[23] A. B. Downey, "Using pathchar to estimate Internet link characteristics," *ACM SIGCOMM*, pp. 241–250, 1999.

[24] S. Sargento and R. Valadas, "Accurate estimation of capacities and cross-traffic of all links in a path using ICMP timestamps," *Telecommunications Systems*, Vol. 33, No. 1–3, pp. 89–115, 2006.

[25] Y. Labit, P. Owezarski and N. Larrieu, "Evaluation of active measurement tools for bandwidth estimation in real environment," *IEEE/IFIP E2EMON*, pp. 71–85, 2005.

[26] T. Goto, A. Tagami, T. Hasegawa and S. Ano, "TCP throughput estimation by lightweight variable packet size probing in CDMA2000 1x EV-DO network," *IEEE/IPSJ SAINT*, pp. 1–8, 2009.

[27] Akamai Technologies, Inc., "The state of the Internet, 2nd quarter, 2009," Vol. 2, No. 2, October 2009, [online] http://www.akamai.com/html/about/press/releases/2009/press_100109.html

[28] K. Yoshida et al., "Inferring POP-level ISP topology through end-to-end delay measurement," *PAM Conference*, pp. 35–44, 2009.

[29] ITU-T recommendation G.1010, "End-user multimedia QoS categories," November 2001.

[30] R. Steinmetz, "Human perception of jitter and media synchronization," *IEEE JSAC*, Vol. 14, Issue 1, pp. 61–72, 1996.