JitterPath: Probing Noise Resilient One-Way Delay Jitter-Based Available Bandwidth Estimation

Yu-Chen Huang, Chun-Shien Lu, *Member, IEEE*, and Hsiao-Kuang Wu

***Abstract*—Measurement of end-to-end available bandwidth has received considerable attention due to its potential use in improving QoS. Available bandwidth enables the sending rate to adapt to network conditions, so that packet loss, caused by congestion, can be significantly reduced before error control mechanisms are finally employed. To this end, we propose a probing noise resilient available bandwidth estimation scheme, called JitterPath, which is adaptive to both the fluid and bursty traffic models. Two key factors, one-way delay jitter and accumulated queuing delay, are both exploited to predict the type of queuing region for each packet pair. Then, the bottleneck utilization information included in the joint queuing regions is estimated and used to quantify the captured traffic ratio, which indicates the relationship between the probing rate and available bandwidth. The contributions of our method are as follows: 1) JitterPath can work without being restricted to fluid traffic models; 2) since JitterPath does not directly use the bottleneck link capacity to calculate the available bandwidth, it is feasible for use in a multihop environment with a single bottleneck; and 3) JitterPath inherently reduces the impact of probing noises under the bursty cross traffic model. Extensive simulations, Internet experiments, and comparisons with other methods were conducted to verify the effectiveness of our method under both single-hop and multihop environments.**

***Index Terms*—Available bandwidth, bottleneck, congestion, one-way-delay jitter, probing, QoS.**

I. Introduction

A. Background

FOR multimedia transmission over a network, bandwidth is a key factor that affects the packet loss rate, which may be severe if the sending rate is not adaptively adjusted according to the network conditions. In order to exploit network bandwidth efficiently, multimedia data (e.g., video) needs to be compressed in advance before transmission. However, when packet loss occurs, reference errors that occur during decoding will be propagated, further degrading the quality of the decoded data. In order to enable decoded video to be displayed with uniform quality, the network needs to provide quality-of-service (QoS) [32] for multimedia transmission. In this paper, we shall focus on available bandwidth estimation, which is closely related to congestion control and adaptive rate adjustment.

In the literature, numerous approaches have been presented for estimating network bandwidth. In [2], Dutta and Zhang improved the current Internet infrastructure so as to support different kinds of QoS in core networks. However, their method undesirably changes the intermediate nodes of a network; thus, scalability is lost. In [15], [16], [25], [32], methods were proposed for adjusting the transmission rate according to the current network conditions. Their common characteristic is that a scalable video coding scheme is incorporated with a network behavior monitor. The key to these solutions lies in the methods used to reliably detect network conditions, where congestion plays a crucial role. Traditional transport protocols (e.g., TCP), however, are still insufficient for dealing with congestion because they need an indication of the packet loss[1] to adjust the sending rate in a blind manner so that video transmission with stable quality cannot be guaranteed. For this reason, the user datagram protocol (UDP) and an additional congestion control mechanism are usually combined to facilitate video transmission. In [33], Yoma *et al.* proposed a new real-time protocol that uses more bandwidth than the TCP-friendly protocols, but does not require any network admission control mechanism to protect the TCP traffic from unacceptable degradation due to the increase of the bandwidth required by UDP applications. Their protocol adapts the bandwidth allocated to a real-time application with the Least Mean Square (LMS) algorithm that aims at reducing the error between packet loss and a desired response. Basically, Yoma *et al.*'s scheme can be seen as a compromise between the conservative TCP-friendly protocols and the ordinary UDP open-loop scheme.

In this paper, in order to deal with congestion, we prefer to control the sending rate through available bandwidth estimation. Our method is based on the fact that bandwidth utilization and variation are closely related to the sending rate, congestion, and packet loss. Our idea is also supported by the findings that

Manuscript received June 14, 2006; revised October 31, 2006. This work was supported by the National Science Council under Grants 93-2213-E-001-026 and 94-2213-E-001-027. Partial results of this study were published in [8]. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Anna Hac.

Y.-C. Huang is with the Institute of Information Science, Academia Sinica, Taipei 115, Taiwan, R.O.C. and is also with the Department of Computer Science and Information Engineering, National Central University, Chung-Li, Taiwan, R.O.C..

C.-S. Lu is with the Institute of Information Science, Academia Sinica, Taipei 115, Taiwan, R.O.C. (e-mail: lcs@iis.sinica.edu.tw).

H.-K. Wu is with the Department of Computer Science and Information Engineering, National Central University, Chung-Li, Taiwan, R.O.C..

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2007.893343

[11] In this work, we focus on the use of end-to-end statistics for available bandwidth estimation. Therefore, we do not consider Explicit Congestion Notification (ECN) because it needs intermediate node support in the bottleneck router. In addition, a new protocol, called DCCP, is designed for congestion control of data that is sent via UDP, but is not considered here because its response time is long. As for FreeBSD, it uses bandwidth delay product to adjust window size to achieve a fair share of bottleneck bandwidth. This idea is similar to TCP-Westwood. The key is how to estimate fair bandwidth estimation. However, fair bandwidth estimation needs to use filtering for measurement noise elimination, which is still challenging.

1) the aggregated TCP throughput is stationary over a long-term period of time, and the individual TCP rate varies by less than a factor of hours [5], [34] and 2) intermediate nodes on the Internet are often stable, and the routing paths of flows remain unchanged for long periods of hours or days [5], [23]. The above facts indicate that available stable bandwidth exists and can be used to predict the future behavior of a propagation path over the Internet. In addition, available bandwidth can be used to improve the performance of a network. For example, Hu and Steenkiste proposed using *PaSt* [4] to set the slow-start threshold and *Pathneck* [6] to detect the location of a bottleneck based on active estimation of the available bandwidth so that the bandwidth could be sufficiently utilized, while Ribeiro *et al.* [28] proposed locating available bandwidth bottlenecks through available bandwidth estimation.

Active probing has been found to be useful for discovering network conditions, including available bandwidth. In [13], Keshav proposed using a new idea, called the "packet pair," to estimate the capacity of a bottleneck link (or narrow link), which is defined as the minimum capacity among the links between the sender and the receiver. However, the capacity of a bottleneck link is not really significant enough to affect network conditions. On the other hand, available bandwidth [3], [30], which is determined from all the links as the minimum bandwidth that has not been used [10], has been recognized as the most important factor. Formally, the link with minimum available bandwidth is referred to as tight link, which is the bottleneck of an underlying propagation path. Thus, it is known that a bottleneck link is not necessarily a tight link in a multihop network model.

B. Related Works

In [30], active probing-based available bandwidth estimation technologies were roughly divided into two categories: 1) probe gap model (PGM) methods and 2) probe rate model (PRM) methods. In this paradigm, a probing train, which is composed of several packet pairs, is sent to collect end-to-end statistics for available bandwidth estimation. Some representative approaches are reviewed here to facilitate describing our method and comparing it with others later.

1) Probe Rate Model (PRM)-Based Methods: The idea behind PRM-based methods [7], [11], [17], [19], [27] is to exploit self-induced congestion, which relies on an intuitive assumption that if the probing rate is higher than the available bandwidth, then the probing packets will start queuing in the bottleneck router, resulting in increasing one-way delay. More specifically, one-way delays are determined by three factors: service, queuing, and transmission delays, where the transmission delay is also the propagation delay and the service delay is defined as the packet size over sending rate. In a single-hop network environment, the one-way delay of the ith packet P^i, OWD^i, at the hop is

$$OWD^i = \tau^i - \tau_*^i = D^i + x^i + t^i \tag{1}$$

where τ_*^i denotes the departure time of P^i sent from the sender, τ^i denotes the arrival time of P^i received at the receiver, D^i denotes the queuing delay, x^i denotes the service delay, and t^i denotes the transmission delay. Based on (1), the OWD jitter, ΔOWD^i, measured as the difference between two one-way delays, can be derived as

$$\begin{aligned}\Delta OWD^i &= OWD^i - OWD^{i-1}\\ &= (t^i - t^{i-1}) + (x^i - x^{i-1}) + (D^i - D^{i-1})\end{aligned} \tag{2}$$

which can be further simplified as

$$\Delta OWD^i = D^i - D^{i-1} \tag{3}$$

based on the assumptions that the difference between transmission delays can be eliminated if the packets are transferred over the same path and that the difference between service delays can be eliminated if the packet size is the same.

Similarly, we consider a multihop with a single bottleneck environment. Let the propagation path be composed of H links, and let each hop, hop_h $(1 \leq h \leq H)$, consist of a FIFO queue and a store and forward router. The OWD jitter of the multihop path can be obtained by summing the OWD jitters (3) of all the hops for the ith packet pair, which is simplified due to domination of the tight link as

$$\Delta OWD^i = D_{tl}^i - D_{tl}^{i-1} \tag{4}$$

where $tl(\in [1\ H])$ denotes the location of the tight link and $D_h^i - D_h^{i-1} = 0$ holds for $h \neq tl$. The OWD jitters in (4) can result in two trends: increasing OWD and decreasing OWD.

The probability of OWD increasing trend as adopted in [17] is determined by comparing[2] two OWDs at a hop as follows:

$$Prob\{OWD^l > OWD^m\} > 0.5,\ \forall\, l > m. \tag{5}$$

Equation (5) implies that if the probability of an increasing OWD ratio is larger than 0.5, then increasing OWD will occur in an OWD sequence. In addition, (5) also implies that a bottleneck will occur no matter whether it is located in the tight link or bottleneck link. This is the unique characteristic of PRM-based methods.

2) Probe Gate Model (PGM)-Based Methods: In PGM-based methods [3], [21], [26], [30], probing packets are sent into the network in the hope that the inter-departure time can be different from the inter-arrival time at the receiver side. This information can be exploited to measure the average cross traffic rate. Under this circumstance, it is possible to directly estimate the available bandwidth as the bottleneck link capacity minus the cross traffic rate. Two assumptions are made in PGM so that reliable and fast available bandwidth estimation can be better guaranteed without the need to pour many probing trains into the network. They are 1) a single-hop environment is assumed such that the link with the minimum available bandwidth is the same as the bottleneck link and 2) the fluid cross traffic model is assumed so that probing noises[3] will not exist and the packet pairs in a packet train will operate in the same queuing region. Two kinds of queuing regions are used to indicate the load of a bottleneck: a joint queuing region (JQR)

[22]In [11], a different type of comparison was adopted.

[33] Probing noise is a kind of vacant space within a packet pair, affecting the correctness of queuing region classification. We will discuss it later in Section IV.

indicates that the bottleneck is congested during the inter-arrival time when a packet pair reaches the receiver; a disjoint queuing region (DQR) indicates that the bottleneck is not congested.

Let us take IGI [3] as an example to illustrate how a PGM-based method works. Basically, IGI involves two tasks: determining the queuing region where a packet train operates and estimating the cross traffic rate. For queuing region determination, it is said that a packet train is operating in a joint queuing region (JQR) if

$$\frac{\left|\sum_{i=1}^{n} \Delta_{j,out}^{i} - n \times \Delta_{j,in}\right|}{max\left(\sum_{i=1}^{n} \Delta_{j,out}^{i}, n \times \Delta_{j,in}\right)} > \delta \tag{6}$$

holds, where n is the number of packet pairs in a packet train, $\Delta_{j,in}$ denotes the inter-departure time (or input probing gap) of the jth packet train at the sender, $\Delta_{j,out}^{i}$ denotes the inter-arrival time (or output probing gap) of the ith packet pair of the jth packet train arriving at the receiver, and δ is the threshold used to determine the queuing region type of a packet train. In [3], Hu and Steenkiste suggested that δ be set to 0.1 experimentally. They also assumed use of the fluid cross traffic model with a constant rate.

When the packet pairs operate in a JQR, the inter-departure time of a packet pair is simply assumed to be fully filled with cross traffic. As a result, the cross traffic rate, λ, with respect to the jth packet train can be derived as

$$\lambda = \frac{\sum_{i=1}^{n} \left(\Delta_{j,out}^{i} \times TC - s\right)\Big|_{\text{if } \Delta_{j,out}^{i} > \Delta_{j,in}}}{n \times \Delta_{j,in}} \tag{7}$$

where TC denotes the tight link capacity and the constant value, s, denotes the probing packet size. Since IGI assumes operation in a single-hop environment, the bottleneck link capacity (BC) is equal to the tight link capacity (TC). Therefore, the available bandwidth, $Avbw$, can be directly estimated as $Avbw = BC - \lambda$. However, in a multihop environment, it is known that $BC \leq TC$.

C. Our Contributions

Since this study focuses on the use of end-to-end statistics for available bandwidth estimation, we propose a new method, called JitterPath, based on the exploitation of one-way delay jitter. Among various end-to-end statistics, our method is based on one-way delay jitter because: 1) The use of packet loss is feasible only if the distinction between congestion loss and wireless loss is performed, which is still challenging and not the main scope of this paper. 2) The use of round-trip-time (RTT), whose accuracy is affected by the queuing delay of the feedback channel, is not reliable. 3) As one-way-delay (OWD) is composed of the service time, propagation time, and queuing delay, if the packets are of the same size and transmitted over the same path, then OWD jitter will be composed of queuing delay jitter only, which can provide accurate measurement information.

JitterPath includes a strategy called the "queuing delay propagation model (QDPM)" that can be used to predict whether the bottleneck is congested during the measured period, and can inherently alleviate the probing noises caused in the bursty cross traffic model. Our method also defines the "captured traffic ratio (CTR)," which is the ratio of the total output gaps accumulated from JQRs to the total input gaps and is used to determine the relationship between the probing rate and available bandwidth. Thus, CTR indicates whether a bottleneck is congested or not during the measured time, which is the total inter-arrival time of the packet pairs in a packet train. In summary, there are three major contributions of our method: 1) JitterPath can work without assuming the use of only a fluid traffic model; 2) since JitterPath does not directly use a bottleneck link capacity to calculate the available bandwidth, it is feasible for use in a multihop environment with a single bottleneck; and 3) JitterPath can alleviate the impact of probing noises under the bursty cross traffic model. A comparison among the PGM methods, PRM methods, and the proposed method, is shown in Table I.

TABLE I
COMPARISON AMONG DIFFERENT TYPES OF AVAILABLE BANDWIDTH ESTIMATION METHODS

Method	Single-hop only	Fluid cross traffic only
PGM [3], [21], [26], [30]	Yes	Yes
PRM [7], [11], [17], [19], [27]	No	Yes
JittherPath (this work)	No	No

The remainder of this paper is organized as follows. In Section II, we point out the key drawbacks of the existing PRM- and PGM-based available bandwidth estimation methods and describe our strategies. In Section III, we first describe the proposed method, JitterPath, which contains a so-called "queuing delay propagation model (QDPM)" to precisely determine the type of queuing region for each packet pair and uses a so-called "captured traffic ratio" to indicate the relationship between the probing rate and available bandwidth under the fluid traffic model. In Section IV, our method, JitterPath, is presented based on the bursty traffic model, which considers the impact of probing noise when measuring the available bandwidth. In Section V, simulation and real-network results are given and comparisons with other methods are made to demonstrate the effectiveness of our method. Finally, concluding remarks are made in Section VI.

II. Problem Description and Our Solution

From a review of the literature, we find that the main drawback of the PRM-based methods is that the accuracy in determining the OWD increasing trend depends on the nature of the cross traffic model. If the cross traffic operates like a fluid model[4] [12], [29], then its arrival rates are constant, and the amount of cross traffic inserted within each packet pair is the same. Under these circumstances, the queuing delays of probing packets will constitute an increasing and smoothing sequence if the probing rate is higher than the available bandwidth. On the other hand, if the probing rate is lower than or equal to the available bandwidth, no queuing delay will occur.

However, TCP traffic, the main traffic over the networks, will affect the accuracy of PRM-based methods. The large-scale

[44] The formal definition of the fluid model is defined as follows. Let λ be the cross traffic rate. During any time interval of length t, the amount of arriving cross traffic at a link is $\lambda \times t$. Thus, it is known that CBR traffic more approximates a fluid model than TCP traffic because the variance of TCP's sending rate is larger than that of CBR's sending rate over the same observed period.

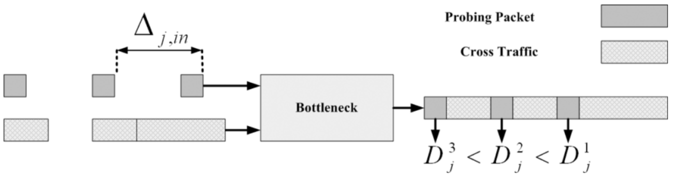

Fig. 1. Estimation error in PRM, where the probing rate $s/\Delta_{j,in}$ of the jth packet train with packets of size s is higher than the available bandwidth, and the cross traffic rate decreases progressively during measurement. "D_j^i" denotes the queuing delay of the ith probing packet. The OWD increasing trend is not detected so as to wrongly determine that the bottleneck is not congested.

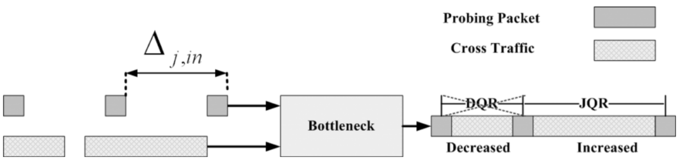

Fig. 2. Error in PGM, where the probing rate is higher than the available bandwidth, and the cross traffic rate decreases progressively during measurement. The decreasing gap is, in fact, a JQR, but IGI incorrectly determines it to be a DQR.

Internet measurement in [5] revealed that aggregated TCP's throughput is stationary if a large number of TCP connections compete for bandwidth with each other over a long-term period of time. Unfortunately, if the observed period is not long enough, TCP traffic will act bursty instead of fluid so the OWD increasing trend does not suffice to represent the relationship between the probing rate and available bandwidth (5), especially when the cross traffic rate varies during measurement. A typical scenario is depicted in Fig. 1, where the OWD increasing trend is not detected even when the probing rate is higher than the available bandwidth.

In view of this fact, the restriction of using fluid traffic only is removed in our method. In addition, the common belief that the OWD decreasing trend is useless for estimating the available bandwidth is also ignored in this study. On the contrary, we find that it does indeed provide useful information for indicating the relationship between the probing rate and available bandwidth. On the other hand, we also show in (5) that only one-way delay jitter is employed to infer the relationship between the probing rate and available bandwidth in the PRM-based methods. As we will show later, this is not sufficient.

The PGM-based methods typically assume the use of the fluid cross traffic model since it can judge that the expanded output gaps are filled with cross traffic, while the reduced output gaps are only caused by previously expanded output gaps. However, the threshold δ in (6), used to judge whether or not a probing train operates in a joint queuing region, is easily affected by the probing noise (to be discussed in Section IV) when bursty cross traffic is encountered. If this threshold cannot be accurately set, then the type of queuing region in which a packet train operates will be wrongly determined, leading to inaccurate estimation of the available bandwidth. A typical scenario is depicted in Fig. 2, where a reduced output gap that is a JQR is wrongly determined as being a DQR due to the assumption that the fluid traffic model is used.

In addition, the PGM-based methods rely on the use of the bottleneck link capacity to directly calculate the available bandwidth. Thus, accurate estimation of the bottleneck link capacity is indispensable. In our method, the bottleneck link capacity is only employed to approximately set the upper bound of the available bandwidth during the process of iterative approximation in available bandwidth estimation. Thus, the BC mainly affects the convergence time instead of the accuracy in our method.

To address the aforementioned problems, we propose a new queuing region determination technique based on one-way delay jitter and queuing delay, where sufficient information is collected to infer the relationship between the probing rate and available bandwidth, i.e., bottleneck utilization. One advantage is that only information about the queuing status of the bottleneck is collected no matter where the bottleneck is located. Another advantage is that a binary search-based strategy is used for iterative available bandwidth estimation to avoid the explicit use of bottleneck link capacity and to eliminate the probing noise. More importantly, since the proposed queuing delay propagation model can efficiently eliminate the probing noise caused by bursty traffic, our method is not limited to the fluid traffic model. All the technical components will be described in the following two sections with respect to the fluid and bursty cross traffic model, respectively.

III. Proposed Method, JitterPath, Based on the Fluid Traffic Model

In this section, the proposed method based on the fluid cross traffic model, in which no probing noise exists and the packet pairs of the same packet train operate in the same queuing region, will be presented first. In Section III-A, a strategy for queuing delay propagation model (QDPM)-based queuing region determination will be presented. It can be used to classify the output gaps of a packet train at the receiver side. In addition, the queuing delay is updated once a packet pair has been processed by the router. In Section III-B, we will exploit the types of queuing regions and the magnitude relationship between the total output gaps accumulated in JQRs and the total input gaps to investigate the relationship between the probing rate and available bandwidth. Then, in Section III-C, we will describe an iterative probing rate adjustment strategy, where the updated probing rate can gradually approximate the available bandwidth.

A. Queuing Delay Propagation Model (QDPM)-Based Queuing Region Determination

In active probing-based available bandwidth estimation, probing packet trains are sent and interact with cross traffic so that useful information can be collected to derive the available bandwidth. Here, a packet train is composed of several back-by-back packet pairs, and the time interval within a packet pair is called the "probing gap." In other words, the "input probing gap" is defined as the inter-departure time of a packet pair sent from the sender, while the "output probing gap" is defined as the inter-arrival time of a packet pair reaching the receiver. The output probing gap may be different from its corresponding input probing gap because the sent packet train may be affected by cross traffic since the time interval within a packet pair is either compressed or expanded.

The utilization of the bottleneck router during the inter-arrival time of a packet pair is expressed by two different queuing regions (QRs) [3]: a joint queuing region (JQR) and a disjoint queuing region (DQR). More specifically, a JQR is strictly defined as the situation where the bottleneck router is fully busy servicing aggregated traffic during the time interval within a packet pair, which also means that the bottleneck router is never idle during that interval. The aggregated traffic is composed of the probing traffic and cross traffic. Below, a more accurate QR determination method is described by exploiting both one-way delay jitter and queuing delay.

1) Queuing Region Determination: Let a probing packet train be composed of n packet pairs. Let $PP_j^i (1 \leq i \leq n)$, which is composed of consecutive probing packets P_j^{i-1} and P_j^i, denote the ith packet pair in the jth packet train, whose inter-departure time is denoted by $\Delta_{j,in}$. Let QR_j^i denote the queuing region in which PP_j^i operates. If PP_j^i is defined as operating in a JQR, then the bottleneck router does not finish the following three tasks before the arrival of P_j^i: processing the queued traffic (let Q_j^{i-1} denote the sum of the queued traffic and undelivered packets when P_j^{i-1} arrives at the router), processing P_j^{i-1} (let s be the constant packet size), and processing the cross traffic inserted within this packet pair PP_j^i (let CT_j^i denote the amount of cross traffic arriving at the bottleneck during the inter-arrival time of PP_j^i). These tasks can be related to each other as follows:

$$\left(\frac{Q_j^{i-1} + s + CT_j^i}{TC}\right) > \Delta_{j,in}, \quad \text{if } QR_j^i \text{ is a JQR} \qquad (8)$$

where TC denotes the capacity of a tight link. Based on the definition of a JQR, the inter-arrival time of a packet pair PP_j^i measured at the receiver side is $(s + CT_j^i)/TC$, which is the output gap, $\Delta_{j,out}^i$. In addition, let Q_j^{i-1}/TC be denoted as D_j^{i-1}, representing the queuing delay that accumulates before packet P_j^{i-1}. Thus, (8) can be rewritten as

$$\Delta_{j,out}^i > \left(\Delta_{j,in} - D_j^{i-1}\right), \quad \text{if } QR_j^i \text{ is a JQR.} \qquad (9)$$

Equation (9) can be further rewritten by substituting the one-way delay jitter for the difference between the output and input gaps as follows:

$$\Delta OWD_j^i + D_j^{i-1} > 0, \quad \text{if } QR_j^i \text{ is a JQR,} \qquad (10)$$

where $\Delta OWD_j^i = \Delta_{j,out}^i - \Delta_{j,in}$. In fact, (10) shows what can be derived if a packet pair operates in a JQR.

In practice, in order to determine the queuing region type of a packet pair, what the receiver needs based on (10) is the OWD jitter and the accumulated queuing delay. In addition, OWD jitter is also known to closely depend on the accumulated queuing delay, which implies that an accurate queuing delay propagation mechanism is indispensable. We shall discuss this issue in Section III-A2. Thus, the condition that can be used to determine whether a queuing region is a JQR is defined as

$$QR_j^i \text{ is JQR}, \quad \text{if } \Delta OWD_j^i + D_j^{i-1} > 0 \qquad (11)$$

under the fluid traffic model.

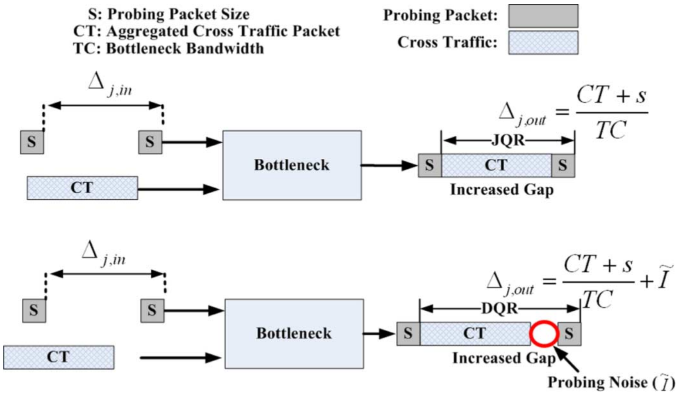

Fig. 3. Appearance of probing noise, which are gaps that are not filled with any traffic data: (upper) the first scenario and (bottom) the second scenario.

In contrast with (5) for the PRM-based methods, our method uses not only OWD jitter but also delay information to classify a queuing region. In addition, $\Delta OWD_j^i + D_j^{i-1} > 0$ in (11) reveals that the packet pair PP_j^i captures a certain amount of cross traffic. However, the queuing region QR_j^i of PP_j^i is not certain to be a JQR because the captured cross traffic is not guaranteed to completely fill the gap within PP_j^i. If $\Delta OWD_j^i + D_j^{i-1} > 0$ and space exists (see Fig. 3) within PP_j^i, then the output gap is actually a DQR but is erroneously determined to be a JQR. This estimation error is caused by the probing noise (i.e., empty gaps) and will be further studied in more detail in Section IV by taking the bursty traffic model into consideration.

2) Queuing Delay Propagation: Once QR_j^i has been determined based on (11), the accumulated queuing delay D_j^i needs to be calculated so that the next queuing region can be determined. Depending on whether packet pair PP_j^i operates in a JQR or not, the queuing delay that accumulates in front of packet P_j^i can be derived as

$$D_j^i = \begin{cases} D_j^{i-1} + \Delta OWD_j^i, & \text{if } QR_j^i \text{ is JQR} \\ max\left(0, \Delta OWD_j^i\right), & \text{if } QR_j^i \text{ is DQR.} \end{cases} \qquad (12)$$

As a result, if the procedures shown in (11) and (12) are iteratively performed, then the queuing region determination and queuing delay propagation can be accomplished. However, it should be noted that D_j^0, the "initial queuing delay," is the queuing delay in front of the first probing packet in the jth packet train and is still unknown. This will be addressed in the next subsection.

3) Determining the Initial Queuing Delay: Since the accumulated queuing delay and one-way delay jitter play key roles in determining the type of queuing region, they are exploited here to determine the initial queuing delay. Our method only needs to trace the output gaps of a packet train once to determine the initial queuing delay. We will investigate this issue based on two cases.

First, we will consider the case where the first OWD jitter is less than zero, i.e., $\Delta OWD_j^1 < 0$. This means that the first output gap is reduced to absorb the initial queuing delay. The size of the first output gap depends on two factors: the queuing delay and cross traffic. If there is no cross traffic or if the amount of inserted cross traffic is not large enough to expand the first output gap, then $|\Delta OWD_j^1|$ is equal to the initial queuing delay D_j^0. On the other hand, if $D_j^0 > |\Delta OWD_j^1|$, then subsequent output probing gaps will be gradually reduced so as to absorb D_j^0. Propagation of the queuing delay will continue, and a decreasing sequence of queuing delays, $D_j^0 > D_j^1 > \cdots > D_j^{k-1}$, will be generated until $D_j^{k-1} < D_j^k$, which is caused by the inserted cross traffic, is satisfied. This means that the initial queuing delay will be completely exhausted. Therefore, the initial queuing delay can be calculated and found to be the absolute value of D_j^{k-1}, $|D_j^{k-1}|$.

In the second case, the first OWD jitter is larger than zero, i.e., $\Delta OWD_j^1 > 0$. In this situation, we cannot be sure whether the initial queuing delay will be completely absorbed in the first output gap or if it does not exist initially. However, we can infer the queuing delay D_j^1 of the second packet based on (12) and use it as though it were the initial queuing delay, even though we will lose the aggregated traffic captured in the first output gap. This phenomenon might slightly affect the "captured traffic ratio" determination (which will be discussed in the next section). However, its short-term impact on the accuracy of available bandwidth estimation may, in fact, be negligible when compared with the long-term impact of probing noise.

After the queuing regions and queuing delays in a packet train have been determined, the so-called "captured traffic ratio (CTR)" can be quantified to determine the relationship between the probing rate and available bandwidth.

B. *Captured Traffic Ratio Under the Fluid Model*

When a single packet pair is considered, the relationship between the probing rate (R) and available bandwidth $(Avbw)$ can be expressed as

$$R > Avbw \quad \text{iff } \frac{s}{\Delta_{\text{in}}} > \left(TC - \frac{CT}{\Delta_{\text{in}}}\right) \tag{13}$$

where Δ_{in} denotes the input probing gap within the packet pair, $R = s/\Delta_{\text{in}}$ denotes the probing rate of a packet pair, CT/Δ_{in} denotes the cross traffic rate captured by the packet pair, and $TC - (CT/\Delta_{\text{in}})$ denotes $Avbw$. When a packet train is considered, (13) can be rewritten for all packet pairs of the jth probing packet train with probing rate $R_j = s/\Delta_{j,in}$, which is composed of n packet pairs, as

$$R_j > Avbw \quad \text{iff } \sum_{i=1}^{n} \left(CT_j^i + s\right) > n \times TC \times \Delta_{j,in} \tag{14}$$

where CT_j^i denotes the amount of cross traffic captured by packet pair PP_j^i in the jth packet train.

Moreover, the sum of the probing and cross traffic, denoted by $CT_j^i + s$ in (14), can be further classified into two types according to the kind of queuing region in which a packet pair operates. Thus, (14) can be rewritten as shown in (15) at the bottom of the page. However, no clue can be used to reliably measure the amount of cross traffic in a DQR.[5] Consequently, only the cross traffic, which is captured in JQRs, can be exploited to determine the relationship between the probing rate and available bandwidth. Furthermore, under the conditions that the cross traffic is fluid and the probing rate is larger than the available bandwidth, all packet pairs in a packet train will operate in the same queuing region, i.e., JQR. Consequently, (15) can be rewritten based on this prior knowledge as follows:

$$R_j > Avbw, \quad \text{if } \frac{\sum_{i=1}^{n} \left(CT_j^i + s\right)\big|_{\text{if } QR_j^i \text{ is JQR}}}{n \times TC \times \Delta_{\text{in}}^j} > 1. \tag{16}$$

Substituting $\Delta_{\text{out}}^i = (CT_j^i + s)/TC$ into (16), we have

$$R_j > Avbw, \quad \text{if } \frac{\sum_{i=1}^{n} \Delta_{j,out}^i\big|_{\text{if } QR_j^i \text{ is JQR}}}{n \times \Delta_{j,in}} = CTR_j > 1 \tag{17}$$

where CTR_j denotes the value of the "captured traffic ratio (CTR)," which is the ratio of the total output gaps accumulated in JQRs to the total input gaps, and is a measurement of the bottleneck utilization during the jth packet train probing. On the other hand, when $CTR_j \leq 1$, $R_j \leq Avbw$.

The proposed CTR is an important indicator for measuring the bottleneck's status during packet train probing. If the cross traffic's arrival model is fluid, then the bottleneck's status as measured by means of each packet pair will be the same. However, if the cross traffic is bursty, then a single packet pair can only measure the bottleneck's status at one instant. In view of this fact, CTR is employed to indicate the measured average utilization of the bottleneck. There are critical differences between CTR (in our method) and δ (in the PGM-based methods): (1) CTR acts like δ in (6) in that both indicate the relationship between the probing rate and available bandwidth; (2) CTR can be dynamically adjusted depending on the fluctuation of the cross traffic rate, which makes our method more adaptive to dynamic network conditions.

C. *Probing Rate Adjustment*

According to (17), the relationship between the probing rate of the jth packet train and the available bandwidth can be determined. Binary search [11] is then adopted to iteratively approximate the available bandwidth. As we will described in Section IV-A, iterative approximation also facilitates elimination of probing noise. Given the probing rate R_j of the jth

$$R_j > Avbw \quad \text{iff } \frac{\sum_{i=1}^{n} \left(CT_j^i + s\right)\big|_{\text{if } QR_j^i \text{ is JQR}} + \sum_{i=1}^{n} \left(CT_j^i + s\right)\big|_{\text{if } QR_j^i \text{ is } DQR}}{n \times TC \times \Delta_{\text{in}}^j} > 1 \tag{15}$$

[55] As we will describe in Section IV-B, the amount of cross traffic in DQRs can be approximately estimated based on the bottleneck utilization and captured cross traffic in JQRs.

packet train and the obtained CTR_j, the next probing rate can be determined as follows:

$$\begin{aligned} R_{min} &= R_j, \quad \text{if } CTR_j > 1, \\ R_{max} &= R_j, \quad \text{if } CTR_j \le 1, \\ R_{j+1} &= (R_{min} + R_{max})/2 \end{aligned}$$

where R_{min} denotes the upper bound of $Avbw$ and R_{max} denotes the lower bound of $Avbw$. Although no guideline is available for setting the initial values of R_{min} and R_{max}, we recommend that R_{min} be set to the upper bound of the available bandwidth, which is equal to the capacity of the bottleneck link, and that R_{max} be set to the lower bound of the available bandwidth, which is, ideally, zero. It should be noted that R_{min} is merely used to bound the maximum value of $Avbw$. If R_{min} is arbitrarily set to a large value or a known link's capacity (e.g., first output link capacity), the convergence time for available bandwidth estimation will become longer. On the other hand, the bottleneck link capacity can be estimated by means of available methods, such as pathrate [1] and Nettimer [14].

The above probing rate adjustment procedure stops if $|R_{min} - R_{max}| < \gamma$ holds such that the degree of fluctuation of the available bandwidth is bound to γ, which is called the "estimation resolution." When the stop condition is satisfied, R_{max} is adopted as the estimated available bandwidth. In addition, R_{min} is adopted as the next initial probing rate when the available bandwidth procedure restarts.

So far, it can be seen that the proposed method tries to find the available bandwidth by repeatedly pouring traffic into the network. Once the buffer of the router overflows, the incoming data will be dropped, leading to packet loss. Here, we assume that packet loss is caused by congestion, which occurs in a wired environment. Under this circumstance, if the receiver detects probing packet loss, then the sender has to adjust the new probing rate using the AIMD mechanism [9]. However, when the probing rate is reduced to less than R_{max}, this adjustment is not helpful for enabling the probing rate to converge to the available bandwidth. Thus, we propose reducing the next probing rate to $(R_{max} + R_j)/2$ instead of $R_j/2$.

IV. Proposed Method, JitterPath, Under the Bursty Traffic Model

In this section, we will discuss the proposed method under the bursty traffic model. When bursty cross traffic is encountered, the output probing gaps, even those carrying useful information for measuring the cross traffic rate, are contaminated with probing noises, which are defined as gaps that are not occupied by traffic. An example of probing noise is illustrated in Fig. 3, where the aggregated traffic's rate for the two scenarios is larger than the bottleneck capacity. Therefore, a bottleneck will be congested during the inter-arrival time of a packet pair. However, a bottleneck in the second scenario will not be congested due to the appearance of probing noise. Liu *et al.* [18] first pointed out that probing noise is caused by the superposition of incoming probing and cross traffic even if the probing rate is larger than the available bandwidth. Let the probing noise be denoted by $\tilde{I}^i_{\Delta_{j,in}}$. The following relationship can be established [18]:

$$\Delta^i_{j,out} = \frac{CT^i_j}{TC} + \frac{s}{TC} + \tilde{I}^i_{\Delta_{j,in}}. \tag{18}$$

The probing noise, $\tilde{I}^i_{\Delta_{j,in}}$, is the key reason for inaccurate measurement of the cross traffic rate (7) with the PGM-based methods, which exploit the relationship between the output gap and input gap. Since the PGM methods exploit the condition where $\Delta^i_{j,out} = \Delta_{j,in}$ holds to infer the available bandwidth, (18) can be rewritten as

$$TC - \left(\frac{CT^i_j}{\Delta_{j,in}} + R_j\right) = \frac{\tilde{I}^i_{\Delta_{j,in}} \times TC}{\Delta_{j,in}}. \tag{19}$$

In (19), if the probing rate, R_j, is equal to the available bandwidth and if the cross traffic exactly fills the gap within the packet pair PP^i_j (where the queuing region of PP^i_j is a JQR), then the probing noise $\tilde{I}^i_{\Delta_{j,in}}$ must be zero. In addition, if the cross traffic does not completely fill the gap within PP^i_j but creates a false phenomenon such that $\Delta^i_{j,out} = \Delta_{j,in}$ (where the queuing region of PP^i_j is a DQR), then probing noise exists. Under these circumstances, we can define the rules for queuing region determination under the bursty cross traffic model as follows:

$$\begin{aligned} &QR^i_j \text{ is } JQR, \quad \text{if } \tilde{I}^i_{\Delta_{j,in}} = 0; \\ &QR^i_j \text{ is } DQR, \quad \text{if } 0 < \tilde{I}^i_{\Delta_{j,in}} \le \Delta_{j,in}. \end{aligned} \tag{20}$$

In order to reduce the impact of probing noise on queuing region determination, we eliminate the probing noise by means of our queuing delay propagation model, as discussed in the next subsection.

A. Eliminating Probing Noise

In [18], Liu *et al.* investigated the quantification of probing noise, $\tilde{I}^i_{\Delta_{j,in}}$, which is determined by the injected probing packets and accumulated queuing delay as follows:

$$\tilde{I}^i_{\Delta_{j,in}} = max\left(0, I^i_{\Delta_{j,in}} - \frac{s}{TC} - D^{i-1}_j\right) \tag{21}$$

where $I^i_{\Delta_{j,in}}$ is the idle period during which no probing traffic data is serviced, and $0 \le I^i_{\Delta_{j,in}} \le \Delta_{j,in}$. However, they focused on probing-based single hop analysis without presenting a probing noise-resilient available bandwidth estimation method. In the following, we shall analyze two cases to show that the probing noise can be effectively eliminated by means of our queuing region determination strategy. It should be noted that both cases discussed below involve interference caused by probing noises such that a queuing region is erroneously determined as a JQR, while it is actually a DQR.

In the first case, $\Delta OWD^i_j < 0$. In addition, we assume that $\Delta OWD^i_j + D^{i-1}_j > 0$ holds so that the queuing region of PP^i_j is determined as a JQR based on (11), and $D^i_j < D^{i-1}_j$ can be derived based on (12). If probing noise continues to exist, then a sequence of decreasing queuing delays can be built. When the decreasing queuing delay sequence converges to zero, i.e.,

$D_j^k = 0$ (for $k > i$), our method can accurately determine that packet pair PP^{k+1} is a DQR based on (11) since $\Delta OWD_j^{k+1} + D_j^k < 0$. Therefore, the negative impact of probing noise can finally be eliminated.

In the second case, $\Delta OWD_j^i > 0$. Again, we assume that $\Delta OWD_j^i + D_j^{i-1} > 0$ holds so that the queuing region of PP_j^i is determined as a JQR. Based on (12), we can obtain $D_j^i > D_j^{i-1}$. If probing noise continues to exist, then a sequence of increasing queuing delays can be built. In this situation, we can see from (21) that probing noise $\tilde{I}^k_{\Delta_{j,in}}$ (for $k > i$) can be eliminated when the queuing delay D_j^{k-1} is sufficiently large to make $I^k_{\Delta_{j,in}} < (s/TC) + D_j^{k-1}$.

We have found that our queuing delay propagation model is inherently able to stop or eliminate probing noises. Simulation results will be presented later to confirm our analysis that longer probing trains are able to reduce measurement errors. However, probing noises will still remain if the length of a probing packet train is not sufficient. Under this circumstance, the output gaps will be contaminated by probing noises, and CTR_j in (17) will be incorrectly estimated. In the next subsection, we will further investigate how to approximately estimate the residual probing noise and derive the impact of probing noise on the relationship between the probing rate and available bandwidth under the bursty cross traffic model.

B. Captured Traffic Ratio (CTR) Under Bursty Cross Traffic

By summing both sides of (18) and letting $R_j = s/\Delta_{j,in}$, we can derive

$$\frac{\sum_{i=1}^{n} \Delta^i_{j,out}}{n \times \Delta_{j,in}} = \frac{\sum_{i=1}^{n} CT_j^i + n \times R_j}{n \times TC} + \frac{\sum_{i=1}^{n} \tilde{I}^i_{\Delta_{j,in}}}{n \times \Delta_{j,in}} \quad (22)$$

for all packet pairs in the jth probing packet train, where the first term (let $(\sum_{i=1}^{n} CT_j^i + n \times R_j)/(n \times TC)$ be represented by μ_j hereafter) on the right side of (22) denotes the average bottleneck utilization, and the second term (let $\sum_{i=1}^{n} \tilde{I}^i_{\Delta_{j,in}}/(n \times \Delta_{j,in})$ be represented by ϵ_j hereafter) denotes the average probing noise ratio. In addition, the average bottleneck utilization μ_j can be divided into two parts, CTR_j and $uCTR_j$, where CTR_j is defined in (17) and $uCTR_j$ denotes the un-captured traffic ratio in DQRs, which is $\mu_j - CTR_j$. As a result, (22) can be rewritten as

$$\frac{\sum_{i=1}^{n} \Delta^i_{j,out}}{n \times \Delta_{j,in}} = CTR_j + (uCTR_j + \epsilon_j). \quad (23)$$

We further let $\widehat{\epsilon_j}$ denote the sum of $uCTR_j$ and ϵ_j in DQR gaps. As a result, we can derive

$$\widehat{\epsilon_j} = \frac{\sum_{i=1}^{n} \Delta^i_{j,out}}{n \times \Delta_{j,in}} - CTR_j \quad (24)$$

where $0 \leq uCTR_j \leq \widehat{\epsilon_j}$. In this study, $\widehat{\epsilon_j}$ denotes noncaptured traffic ratio (non-CTR) that appears due to probing noise at the jth probing. If probing noise ϵ_j is eliminated, then the originally uncaptured traffic in DQRs will become captured such that $\widehat{\epsilon_j} = 0$.

It is well known that the sum of average cross traffic and probing rate is larger than the tight link capacity or the bottleneck utilization is larger than 1 if and only if the probing rate is larger than the available bandwidth. In other word, $R_j > A$ iff $CTR_j + uCTR_j > 1$. Due to probing noise, $uCTR_j$ cannot be measured. We use a strong condition, $CTR_j > 1$, to infer $R_j > A$. In addition, $CTR_j \leq 1 - \widehat{\epsilon_j}$ $(\leq 1 - uCTR_j)$ is used to infer $R_j < A$. Thus, the relationship between the probing rate and available bandwidth can be derived as

$$\begin{aligned} R_j &> Avbw, \quad \text{if } CTR_j > 1, \\ R_j &\leq Avbw, \quad \text{if } CTR_j \leq 1 - \widehat{\epsilon_j}. \end{aligned} \quad (25)$$

Comparing (17) and (25), we find that the probing noise ratio affects the estimation of the available bandwidth. More importantly, the existence of probing noise may lead to the erroneous classification of a DQR as a JQR, as illustrated in Fig. 3. When $1 - \widehat{\epsilon_j} < CTR_j < 1$ holds, the relationship between probing rate and available bandwidth cannot be determined within this ambiguous region. We will discuss this issue in the next subsection. In addition, we will also investigate the relationship between the noncaptured traffic ratio and additional probing rate that is used for elimination of probing noise.

C. Relationship Between the Noncaptured Traffic Ratio and Additional Probing Rate

An effective way to eliminate probing noise is to pour more probing traffic into the network, as pointed out in [18]. This corresponds to probing longer packet trains, as described in Section IV-A. However, we should be careful not to pour in too much probing traffic so as to avoid unnecessary network collapse. In this section, we shall investigate the relationship between the noncaptured traffic ratio and additional probing rate used for elimination of probing noise. Since the noncaptured traffic ratio, as indicated in (24), can be estimated in practice, we propose to quantify the maximum additional probing rate (corresponding to the maximum noncaptured traffic ratio) in the case where the probing rate is equal to the available bandwidth and probing noise exists.

Suppose that the average cross traffic rate is λ and that the Pareto on/off cross traffic model is considered. In this case, the maximum and minimum cross traffic rates can be represented by $\lambda + \tau_1$ and $\lambda - \tau_2$, respectively. In order to simplify our analysis, $\tau_1 = \tau_2 = \tau$ is assumed. Under these circumstances, if a packet pair captures the cross traffic with a rate of $\lambda + \tau$, then the probing rate is larger than the available bandwidth point during the inter-arrival time of this packet pair in the bottleneck, and its queuing region operates in a JQR. On the other hand, the queuing region of a packet pair, which captures cross traffic with a rate of $\lambda - \tau$, may operate in a DQR or JQR. If this packet pair can receive the propagated queuing delay of the preceding probing packets, then this queuing delay is useful for this packet pair operating in a JQR, as revealed by (20) and (21).

To simplify analysis, we assume that the number of packet pairs, which capture cross traffic with two different rates, are the same and are set to m. In addition, let k $(\leq m)$ denote the number of packet pairs that capture cross traffic with a rate of $\lambda - \tau$ and receive sufficient accumulated queuing delays such that their queuing regions operate in a JQR. Under these circumstances, the CTR of a packet train will be the amount of aggregated traffic in these $m+k$ packet pairs divided by the maximum amount of traffic that the bottleneck can process. The amount of cross traffic, $(\lambda+\tau)\times\Delta_{j,in}$, captured by the m packet pairs and the amount of cross traffic, $(\lambda-\tau)\times\Delta_{j,in}$, captured by the k packet pairs are, respectively, used to substitute for CT_j^i in (15), and after some derivations we get

$$CTR_j = \frac{(m+k)\times TC + (m-k)\times\tau}{2m\times TC}. \tag{26}$$

Moreover, by incorporating (24) and (26), the maximum value of $\widehat{\epsilon_j}$ (which can result in the maximum number of queuing regions determined as DQRs) is found if $k=0$. Substituting $k=0$ into (26), we can derive the maximum $\widehat{\epsilon_j}$ as follows:

$$\widehat{\epsilon_j} = \frac{\sum_{i=1}^{n}\Delta_{j,out}^{i}}{n\times\Delta_{j,in}} - \frac{TC+\tau}{2TC}. \tag{27}$$

In order to reduce the effect of probing noise, the sender needs to use an additional probing rate $\widehat{\omega_j}$, to capture the un-captured traffic, which is composed of probing and cross traffic in those $m-k$ packet pairs operating in DQRs. This implies that our goal is to exhaust the unused bandwidth, such that the aggregated traffic will fill the gaps between packet pairs. To this end, the amount of un-captured traffic is first calculated as $(m-k)\times\Delta_{j,in}\times(R_j+\lambda-\tau)$. Then, we further consider that the probing rate is equal to the available bandwidth, i.e., $R_j = TC-\lambda$, the aim being to consider the bursty cross traffic with the largest rate. Thus, the un-captured traffic is $(m-k)\times\Delta_{j,in}\times(TC-\tau)$. To achieve maximum capture, $k=0$ is set in order to obtain the maximum un-captured traffic, $m\times\Delta_{j,in}\times(TC-\tau)$. In addition, the amount of additionally probed traffic is $2m\times\Delta_{j,in}\times\widehat{\omega_j}$. To achieve effective capture, the following condition must be guaranteed:

$$2m\times\Delta_{j,in}\times\widehat{\omega_j} = m\times\Delta_{j,in}\times(TC-\tau),\ \text{i.e.,}\ \widehat{\omega_j} = \frac{TC-\tau}{2}. \tag{28}$$

Given (27) and (28), the relationship between $\widehat{\omega_j}$ and $\widehat{\epsilon_j}$ can be derived as

$$\widehat{\omega_j} = TC\times\left(\widehat{\epsilon_j}+1-\frac{\sum_{i=1}^{n}\Delta_{j,out}^{i}}{n\times\Delta_{j,in}}\right). \tag{29}$$

However, the tight-link capacity in (29) is unknown, thus, we use BC derived in [1], [14] to approximate TC in our method.

D. Probing Rate Adjustment in the Ambiguous Region

When CTR_j falls between $1-\epsilon_j$ and 1, both R_{min} and R_{max} are kept unchanged and the next probing rate R_{j+1} is adjusted based on the derived $\widehat{\omega_j}$ as $R_{j+1} = min\{R_j+\widehat{\omega_j}, R_{min}\}$. After this kind of adjustment, once the next value of CTR does not fall within the ambiguous region, (25) is used again. Otherwise, after α additional probing we need to check the relationship between $R_{j+\alpha}$ and R_{min}. If they are close enough to each other, the probing procedure stops.

V. Simulation and Real-Network Results

In order to demonstrate the performance of our method, several simulations using ns2 [20] were conducted based on two different network models. Our method was also compared with PathChirp [27] and IGI [3],[6] which are, respectively, typical PRM- and PGM-based mechanisms, in terms of the accuracy of available bandwidth estimation. Moreover, this choice is also based on the fact that PathChirp [27] has been shown to perform better than Pathload [11]. In this study, three different types of cross traffic, that is, constant bit-rate (CBR) traffic, TCP traffic, and Poisson traffic, were used for performance evaluation. Among them, CBR traffic, which approximates the fluid traffic model, is the least bursty traffic, TCP traffic is the most common traffic on the Internet, and cross traffic with a Poisson distribution was adopted here to emulate bursty and memoryless traffic. The default parameters used in ns2 were set for cross traffic except that the packet size was fixed at 1000 bytes.

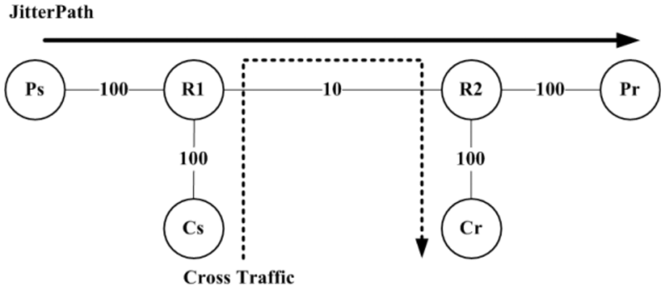

Fig. 4. Single-hop model, where the bottleneck link is the same as the tight link, and the bottleneck bandwidth is set to 10 Mbps. The dashed line denotes the path of cross traffic.

In JitterPath, the value of γ was fixed at 200 Kbps. The initial value of $\widehat{\epsilon_j}$ is set to be γ/BC. The probing rate was initially set to half of the bottleneck link capacity that can be estimated using the methods [1], [14], and its variation during available bandwidth measurement is plotted using dash-dot curves in the figures shown below. For IGI, we adopted a packet train composed of 60 packet pairs (i.e., $n=60$), and the size of each packet was 700 bytes, as suggested in [3], so that the best results could be obtained. For PathChirp, we adopted a packet train with a packet size of 1200 bytes, as suggested in [27]. In our method, a packet train composed of 100 packet pairs (i.e., $n=100$) and each packet of size 1000 bytes were adopted. The actual available bandwidth was averaged for every second by means of the trace log of ns2 simulation.

In addition to extensive simulations, Internet experiments were also conducted to further confirm the performance achievable with our method.

A. Single-Hop Network Environment

The first network topology, as shown in Fig. 4, specifies a single-hop environment, where Ps and Pr denote, respectively, the sender and receiver in the end-to-end probing path, and Cs and Cr denote, respectively, the sender and receiver in the cross traffic transmission path. Three different kinds of cross traffic

[66] In order to estimate the available bandwidth in a multihop environment, the authors of [3] also proposed a PRM-based method, called PTR. PTR uses (6) to detect the state where the probing rate is equal to the available bandwidth. However, the accuracy of available bandwidth determination still depends on the threshold in (6). As shown in (19), this threshold tends to be affected by probing noise under the bursty cross traffic model. In addition, as reported in [3], IGI and PTR achieve comparable performance.

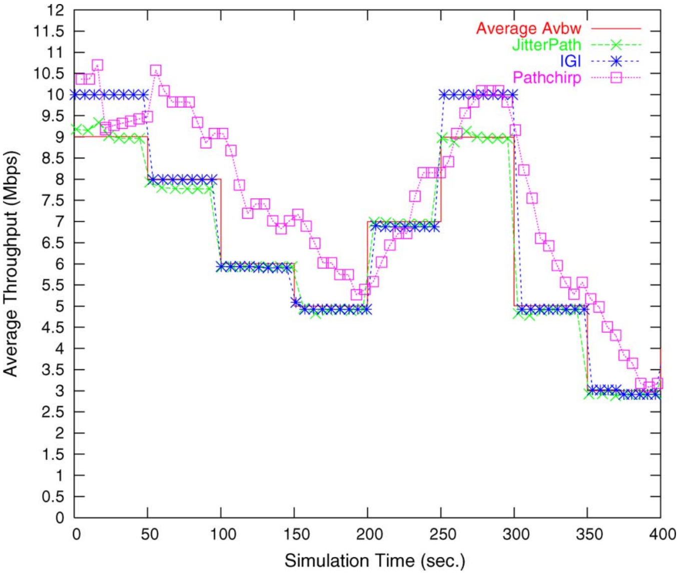

Fig. 5. Comparison of the available bandwidth estimation results obtained with the proposed method (JitterPath), PathChirp, and IGI for the single-hop network model (Fig. 4) with CBR cross traffic only.

were, respectively, used to evaluate available bandwidth estimation. In the first scenario, the cross traffic contained CBR traffic only, which implied that the cross traffic rate was stable. In the second scenario, the cross traffic was composed of TCP traffic only. In this case, we added the first TCP flow at the beginning of network transmission and added a new TCP flow from Cs to Cr every 25 s. In the third scenario, the cross traffic was a Poisson distribution with a rate of 8 Mps, a bursty period of 5 ms, and an idle period of 10 ms. New probing was re-started after available bandwidth is estimated.

The results of available bandwidth estimation obtained using our method, IGI, and PathChirp, and the actual average available bandwidth are plotted in Figs. 5–7 for the above three scenarios.

Numerical comparisons among JitterPath, IGI, and PathChirp in terms of the mean and median of the measurement error, measurement time, and injected bandwidth as well as the mean square error (MSE) of the measurement error in the single-hop model are depicted in Table II. The measurement error is calculated based on the difference between the true and estimated available bandwidth (represented in Kbps). We can observe from this table that our method all obtained measurements with minimal errors. When the cross traffic was TCP or Poisson, the bursty behavior caused IGI and PathChirp to incorrectly estimate the relationship between the probing rate and available bandwidth. However, since our method can properly reduce the probing noise caused by bursty cross traffic, the available bandwidth could be more accurately estimated. On the other hand, when the cross traffic was composed of CBR traffic only, the cross traffic approximated the fluid model, which fits the assumption of IGI, so that IGI could obtain comparable results with ours. Finally, among these three approaches our method spent more measurement time than IGI and PathChirp, while PathChirp needs to inject the maximum amount of probing packets.

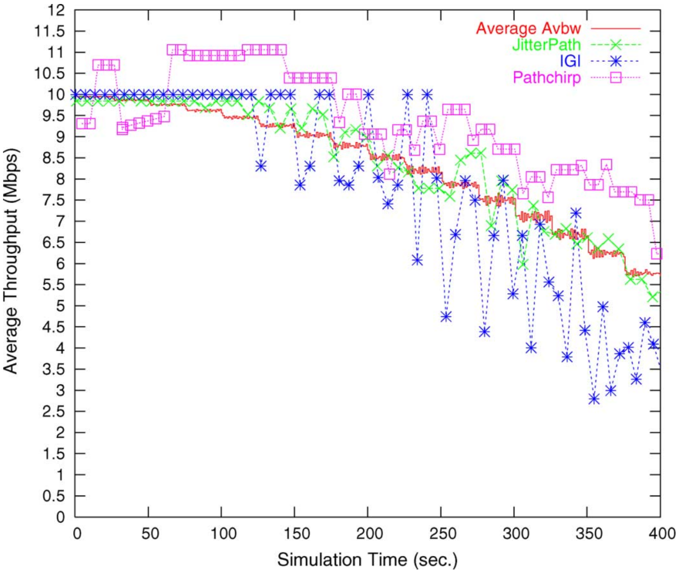

Fig. 6. Comparison of the available bandwidth estimation results obtained with the proposed method (JitterPath), PathChirp, and IGI for the single-hop network model (Fig. 4), where the cross traffic was TCP traffic.

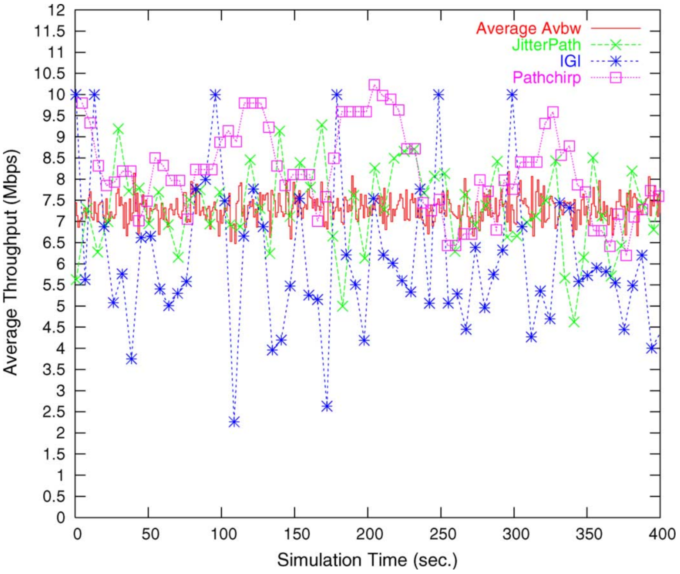

Fig. 7. Comparison of the available bandwidth estimation results obtained with the proposed method (JitterPath), PathChirp, and IGI for the single-hop network model (Fig. 4), where the cross traffic was a Poisson distribution.

TABLE II
NUMERICAL COMPARISON AMONG JITTERPATH, IGI, AND PATHCHIRP IN TERMS OF THE MEAN, MEDIAN, AND MEAN SQUARE ERROR (MSE) OF THE MEASUREMENT ERROR, MEASUREMENT TIME (IN SECOND), AND INJECTED BANDWIDTH (IN Kbps) IN THE SINGLE-HOP MODEL. THE MEASUREMENT ERROR IS CALCULATED BASED ON THE DIFFERENCE BETWEEN THE TRUE AND ESTIMATED AVAILABLE BANDWIDTH (REPRESENTED IN Kbps)

	Method	JitterPath	IGI [3]	PathChirp [27]
Cross Traffic	Metric	(mean, median, MSE)	(mean, median, MSE)	(mean, median, MSE)
CBR	measurement error	(180.22, 77.76, 472.39)	(224.37, 84.16, 446.72)	(1112.57, 1009.21, 1396.07)
	measurement time	(3.03, 3.03, -)	(1.11, 1.04, -)	(0.11, 0.03, -)
	injected bandwidth	(1764.98, 1831.68, -)	(978.91, 982.46, -)	(4893.31, 5929.43, -)
TCP	measurement error	(286.41, 217.01, 365.40)	(1123.04, 842.66, 1500.56)	(1168.92, 1272.67, 1310.71)
	measurement time	(2.22, 2.16, -)	(1.25, 1.66, -)	(0.08, 0.03, -)
	injected bandwidth	(1918.39, 1985.79, -)	(988.06, 1017.15, -)	(5753.82, 6410.17, -)
Poisson	measurement error	(809.20, 740.41, 987.35)	(1736.53, 1691.64, 2018.12)	(1138.57, 976.11, 1386.30)
	measurement time	(1.83, 1.68, -)	(1.36, 1.36, -)	(0.09, 0.05, -)
	injected bandwidth	(1923.37, 1924.47, -)	(1000.39, 1000.68, -)	(3131.27, 2837.27, -)

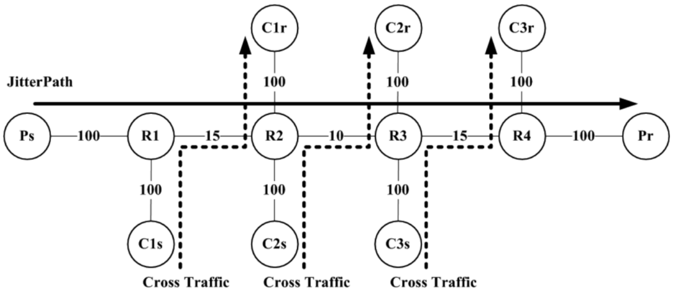

Fig. 8. One hop persistent multihop network model. The bottleneck link capacity (BC) may not be equal to the tight link capacity (TC).

B. Multihop Network Environment

Two different multihop with single bottleneck models were used to evaluate our method and compare its performance with that of IGI and PathChirp.

1) One-Hop Persistent: Fig. 8 shows a one-hop persistent model whose bottleneck can be shifted among all the links. In the network setting, the bandwidth of the first link (R1–R2) was set to 15 Mbps, the bandwidth of the second link (R2–R3) was set to 10 Mbps, and the bandwidth of the third link (R3–R4) was set to 15 Mbps. It should be noted that the available bandwidth of this end-to-end path is the minimum unused bandwidth among the three links. The bottleneck was initially located at bottleneck link (R2–R3) and could be shifted to other links if the cross traffic rate in each link was changed. The initial cross traffic rate in each link was 3 Mbps. New probing was restarted after available bandwidth is estimated.

In this simulation, two bottleneck shifting scenarios were studied. First, the bottleneck was shifted to the link in front of the bottleneck link. This scenario could be achieved by increasing the cross traffic rate of link R1–R2 until link R1–R2 finally became the bottleneck. In the second case, the bottleneck was shifted from link R1–R2 to link R3–R4. This could be achieved by increasing the cross traffic rate of link R3–R4 until the bottleneck was finally transferred to R3–R4. In addition, it should be noted that the bottleneck was shifted from the second link to the first link at the 200th second, and was shifted from the first link to the third link at the 350th second.

In Figs. 9–11, the results for the estimated available bandwidth obtained with our method, PathChirp, and IGI, and the actual average available bandwidth are plotted for comparison purposes. The results shown in Fig. 9 were obtained when the cross traffic was composed of CBR flows only. It can be observed that our method was able to approximate the actual average available bandwidth with slight under-estimations, which were caused by the post narrow effect in that the packet pairs were queued again when the bottleneck is encountered. In fact, this phenomenon exactly appears in the multihop environment. PathChirp yielded rather inaccurate estimations no matter whether the cross traffic rate was high or low. Meanwhile, we found that IGI only estimated the available bandwidth correctly when the bottleneck was exactly located at the bottleneck link. In summary, the major weakness of both PathChirp and IGI as revealed in this simulation is that they tend to incorrectly detect the OWD increasing trend or obtain an incorrect queuing region.

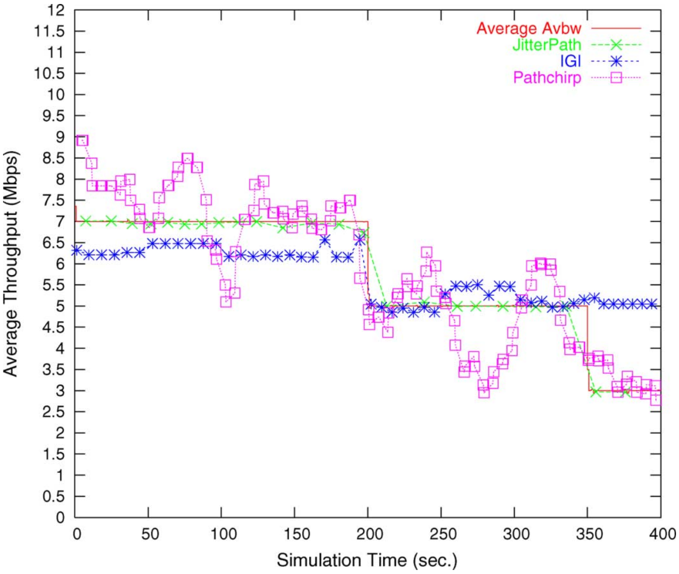

Fig. 9. Comparison of available bandwidth estimations obtained with the proposed method (JitterPath), PathChirp, and IGI for the one-hop persistent model (Fig. 8), where the cross traffic contained CBR traffic only.

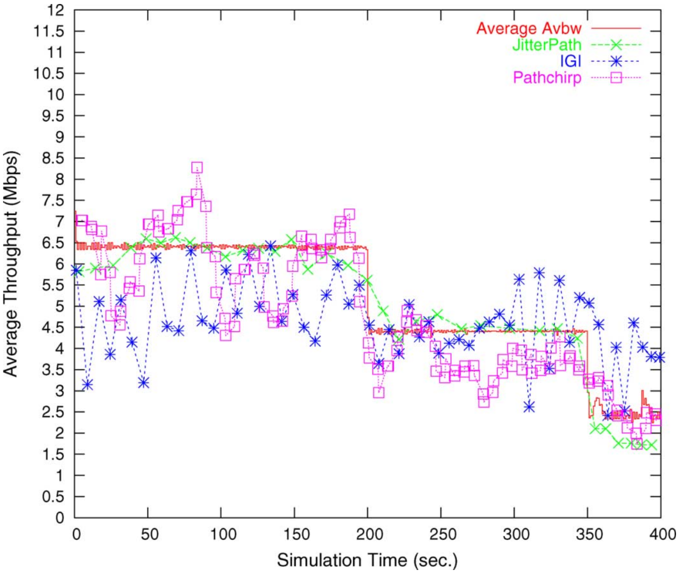

Fig. 10. Comparison of available bandwidth estimations obtained with the proposed method (JitterPath), PathChirp, and IGI for the one-hop persistent model (Fig. 8), where the cross traffic included both CBR and TCP traffic.

The estimation results shown in Fig. 10 were obtained when the cross traffic was composed of both CBR and TCP traffic. The bursty behavior of TCP flows could be simulated by first adding an TCP flow at the beginning of network transmission and adding a second TCP flow at the 200th second. It can be observed from Fig. 10 that both IGI and PathChirp yielded inaccurate and oscillatory estimations. Compared with PathChirp and IGI, our method is capable of obtaining more stable estimations.

The estimation results shown in Fig. 11 were obtained when the cross traffic was composed of both CBR and Poisson traffic. The Poisson traffic had a rate of 3 Mbps, a bursty period of 5 ms, and an idle period of 10 ms. The bursty behavior of Poisson traffic could be simulated by adding three Poisson flows, which

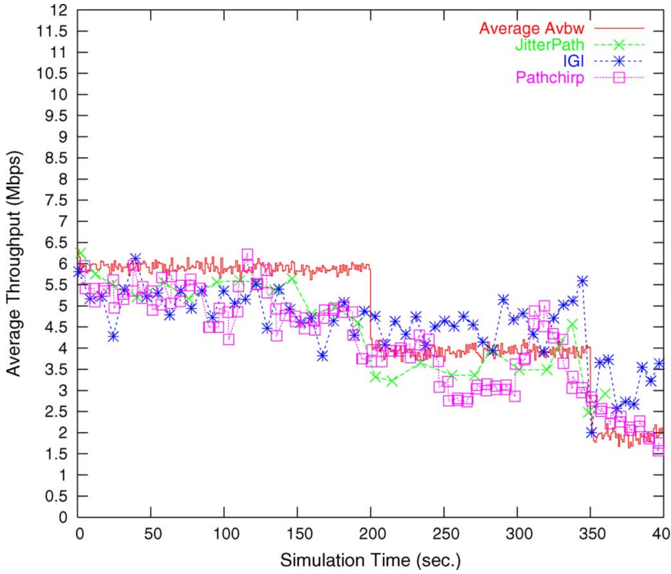

Fig. 11. Comparison of available bandwidth estimations obtained with the proposed method (JitterPath), PathChirp, and IGI for the one-hop persistent model (Fig. 8), where the cross traffic included both CBR and Poisson traffic.

TABLE III
NUMERICAL COMPARISON AMONG JITTERPATH, IGI, AND PATHCHIRP IN TERMS OF THE MEAN, MEDIAN, AND MEAN SQUARE ERROR (MSE) OF THE MEASUREMENT ERROR, MEASUREMENT TIME (IN SECOND), AND INJECTED BANDWIDTH (IN Kbps) IN THE ONE-HOP PERSISTENT ENVIRONMENT. THE MEASUREMENT ERROR IS CALCULATED BASED ON THE DIFFERENCE BETWEEN THE TRUE AND ESTIMATED AVAILABLE BANDWIDTH (REPRESENTED IN Kbps)

	Method	JitterPath	IGI [3]	PathChirp [27]
Cross Traffic	Metric	(mean,median,MSE)	(mean,median,MSE)	(mean,median,MSE)
CBR	Measurement Error	(47.69, 32.67, 73.63)	(680.84, 519.34, 918.25)	(672.49, 536.63, 850.23)
	Measurement Time	(11.87, 9.72, -)	(2.68, 2.24, -)	(0.08, 0.05, -)
	injected Bandwidth	(971.02, 995.20, -)	(458.56, 457.25, -)	(2730.21, 2763.86, -)
CBR+TCP	Measurement Error	(309.38, 202.81, 401.700)	(1072.21, 1025.17, 1363.66)	(712.28, 620.25, 864.44)
	Measurement Time	(6.74, 6.56, -)	(2.10, 2.24, -)	(0.09, 0.05, -)
	injected Bandwidth	(939.23, 958.46, -)	(454.60, 445.81, -)	(2683.44, 2741.99, -)
CBR+Poisson	Measurement Error	(591.50, 487.79, 677.47)	(858.94, 798.24, 977.19)	(699.86, 628.14, 835.23)
	Measurement Time	(9.14, 8.22, -)	(1.93, 2.24, -)	(0.09, 0.05, -)
	injected Bandwidth	(922.51, 945.54, -)	(453.44, 455.81, -)	(2664.24, 2711.86, -)

followed the same paths as CBR flows. We used CBR flows to control the shift of a tight link. It can be observed from Fig. 11 that both IGI and PathChirp yielded inaccurate and oscillatory estimations. Compared with PathChirp and IGI, our method is again capable of obtaining more stable estimations.

Numerical comparisons among JitterPath, IGI, and PathChirp in terms of the mean and median of the measurement error, measurement time, and injected bandwidth as well as the MSE of the measurement error in the one-hop persistent environment are shown in Table III. We can observe from this table that our method consistently obtained the minimum measurement errors. Again, among these three approaches our method spends more measurement time than IGI and PathChirp did, while PathChirp needs to inject the maximum amount of probing packets.

2) Path Persistent: Fig. 12 shows the path persistent model, whose network setting is the same in Fig. 8 except that the bottleneck is still located in the bottleneck link during measurement. Again, the two scenarios and three types of cross traffic used in the one-hop persistent environment were adopted for performance evaluation.

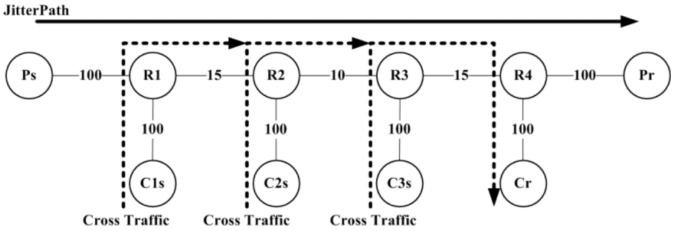

Fig. 12. Path persistent multihop network model. The bottleneck is located in the bottleneck link during measurement.

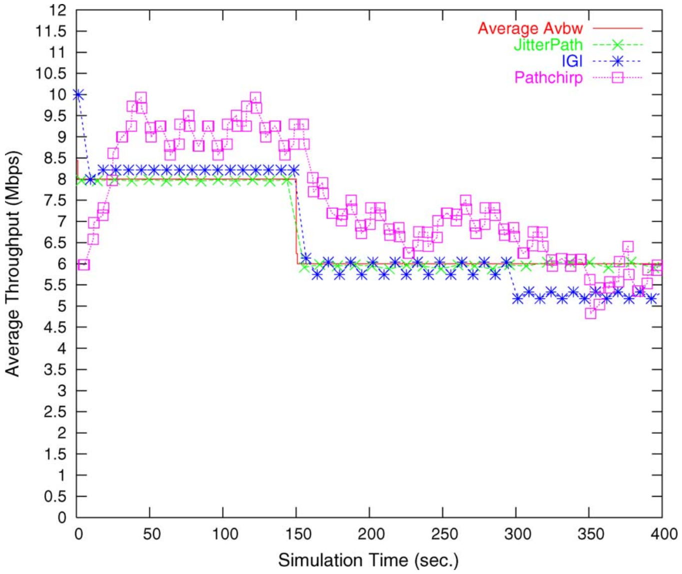

Fig. 13. Comparison of available bandwidth estimations obtained with the proposed method (JitterPath), PathChirp, and IGI for the path persistent model (Fig. 12), where the cross traffic contained CBR traffic only.

In Figs. 13–15, we show the results of available bandwidth estimation obtained from three different sets of cross traffic. In each figure, the results obtained with our method (JitterPath), PathChirp, and IGI, and the actual average available bandwidth are plotted for comparison purposes. From these figures, we can draw conclusions that are similar to those drawn based on the results obtained in the one-hop environment. That is, both IGI and PathChirp yielded inaccurate and oscillatory estimations, while JitterPath again obtained more stable estimations.

Numerical comparisons among JitterPath, IGI, and PathChirp in terms of the mean and median of the measurement error, measurement time, and injected bandwidth as well as the MSE of the measurement error in the path persistent environment are shown in Table IV. Compared with the results obtained in the one-hop persistent environment, it was observed that smaller MSEs can be obtained in the path persistent environment. This is because the bottleneck remained in the bottleneck link, which is similar to the single-hop environment. Among these three approaches, our method spent more measurement time than IGI and PathChirp did, while PathChirp needs to inject the maximum amount of probing packets.

C. Measurement Accuracy versus Packet Train Length

We also examined the relationship between the estimation accuracy achieved with our method and the length of the probing

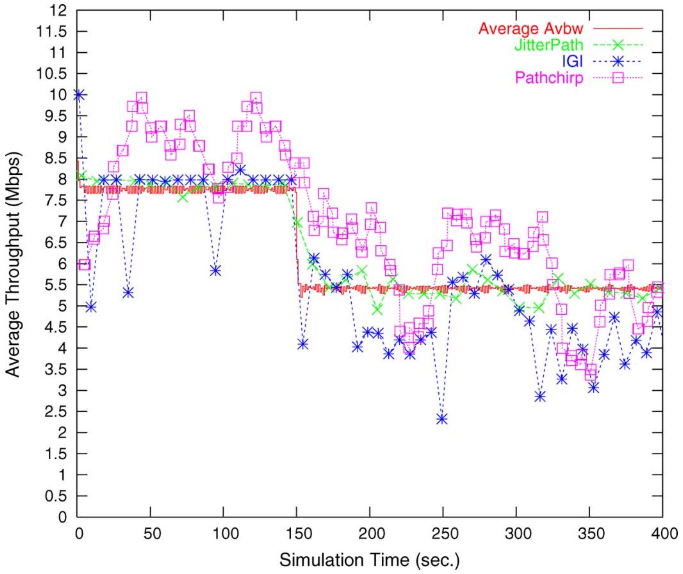

Fig. 14. Comparison of available bandwidth estimations obtained with the proposed method (JitterPath), PathChirp, and IGI for the path persistent model (Fig. 12), where the cross traffic included both CBR and TCP traffic.

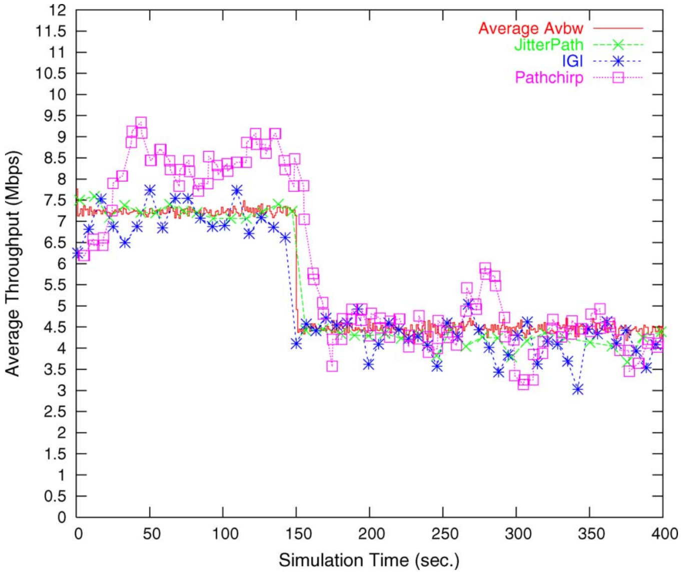

Fig. 15. Comparison of available bandwidth estimations obtained with the proposed method (JitterPath), PathChirp, and IGI for the one-hop persistent model (Fig. 12), where the cross traffic included both CBR and Poisson traffic.

packet train. The single-hop model shown in Fig. 4 was used in this simulation. The obtained numerical results are depicted in Table V. It can be seen that by probing longer packet trains, our method was able to obtain estimations with smaller MSEs, no matter what kind of cross traffic was encountered. This is consistent with our claim in Section IV-A that probing noise can be more efficiently eliminated when a longer packet train is probed.

On the other hand, we can also observe from Table V that as the packet train length became longer and that the measurement error approximately converged to 200 Kbps, which is close to the initial setting of $\gamma = 200$ Kbps, for both CBR and TCP traffic. In the case of Poisson traffic, the measurement error remained about 600 Kbps for longer packet trains. On the whole, our simulation results indicate that for a packet train composed of more than one hundred packet pairs, the measurement error either reached the default resolution or could not be further reduced.

TABLE IV
NUMERICAL COMPARISON AMONG JITTERPATH, IGI, AND PATHCHIRP IN TERMS OF THE MEAN, MEDIAN, AND MEAN SQUARE ERROR (MSE) OF THE MEASUREMENT ERROR, MEASUREMENT TIME (IN SECOND), AND INJECTED BANDWIDTH (IN Kbps) IN THE PATH PERSISTENT ENVIRONMENT. THE MEASUREMENT ERROR IS CALCULATED BASED ON THE DIFFERENCE BETWEEN THE TRUE AND ESTIMATED AVAILABLE BANDWIDTH (REPRESENTED IN Kbps)

	Method	JitterPath	IGI [3]	PathChirp [27]
Cross Traffic	Metric	(mean,median,MSE)	(mean,median,MSE)	(mean,median,MSE)
CBR	Measurement Error	(46.58, 39.87, 55.52)	(492.96, 565.64, 600.39)	(776.27, 722.45, 956.90)
	Measurement Time	(8.24, 7.55, -)	(2.85, 2.96, -)	(0.07, 0.05, -)
	injected Bandwidth	(980.66, 966.69, -)	(460.15, 460.02, -)	(2759.13, 2777.69, -)
CBR+TCP	Measurement Error	(228.75, 212.21, 279.89)	(1019.89, 875.94, 1285.60)	(962.65, 920.30, 1135.10)
	Measurement Time	(6.33, 5.96, -)	(2.59, 2.24, -)	(0.07, 0.05, -)
	injected Bandwidth	(959.11, 946.77, -)	(458.07, 457.16, -)	(2735.66, 2758.33, -)
CBR+Poisson	Measurement Error	(243.12, 241.22, 304.64)	(433.87, 346.60, 550.50)	(705.10, 556.51, 904.71)
	Measurement Time	(6.82, 5.67, -)	(2.11, 2.24, -)	(0.07, 0.05, -)
	injected Bandwidth	(934.14, 913.88, -)	(455.10, 454.02, -)	(2714.38, 2727.27, -)

TABLE V
JITTERPATH: MEASUREMENT ERROR (IN TERMS OF MSE) VERSUS PACKET TRAIN LENGTH IN THE SINGLE-HOP ENVIRONMENT. THE LENGTH OF A PACKET TRAIN DENOTES THE NUMBER OF PACKET PAIRS

Packet train length	60	100	150	200	250
CBR	557.24	286.77	295.59	246.05	273.61
TCP	566.30	461.50	235.99	204.80	213.41
Poisson	963.39	793.36	727.58	632.30	628.58

D. Real-Network Environment

Internet experiments were also conducted to evaluate JitterPath, IGI [3] and PathChirp [27] using PlanetLab nodes [24]. It is known that the most accurate measurements can be acquired by MRTG [22], which, however, needs to access all links and their capacities along the path during measurements. As a result, we chose the relative measurement error, which was also employed in [3], as the criterion for verifying the estimation accuracy of these two approaches. The relative measurement error was defined as

$$\frac{|Avbw^e - throughput_{TCP}|}{BC} \tag{30}$$

where $Avbw^e$ is the estimated available bandwidth, BC is the bottleneck link capacity, and $throughput_{TCP}$ is the bulk data transmission rate, which was measured by Iperf [31].

In the Internet experiments, our measurements were collected from five PlanetLab nodes [24] and six paths among them. The five nodes include UCLA (University of California at Los Angeles), UCB (University of California at Berkeley), BU (Boston University), THU (Tsinghua University, People Republic of China), and AS (Academia Sinica, Taiwan, Republic of China). The six paths chosen for available bandwidth estimation include (p1) from AS to BU; (p2) from AS to UCLA; (p3) from AS to UCB; (p4) from THU to BU; (p5) from THU to UCLA; and (p6) from THU to UCB. The parameter setting here was the same as in the simulations. These measurements

TABLE VI
COMPARISON OF THE MEAN, MEDIAN, AND MSE OF RELATIVE MEASUREMENT ERRORS PRODUCED BY JITTERPATH, IGI, AND PATHCHIRP OVER SIX INTERNET PATHS (SENDER → RECEIVERS)

Path	AS→BU	AS→UCLA	AS→UCB
Metric	(mean,median,MSE)	(mean,median,MSE)	(mean,median,MSE)
JitterPath	(0.448, 0.251, 0.994)	(0.288, 0.140, 0.455)	(1.354, 0.528, 2.778)
IGI	(0.591, 0.322, 1.103)	(0.307, 0.103, 0.895)	(2.780, 0.140, 4.487)
PathChirp	(4.506, 4.542, 4.533)	(4.422, 4.399, 4.464)	(9.847, 9.365, 10.670)

PAth	THU→BU	THU→UCLA	THU→UCB
Metric	(mean,median,MSE)	(mean,median,MSE)	(mean,median,MSE)
JitterPath	(1.335, 1.078, 1.985)	(1.051, 0.854, 1.854)	(2.143, 1.875, 2.824)
IGI	(1.757, 1.341, 2.335)	(1.941, 2.142, 2.028)	(5.144, 4.918, 5.991)
PathChirp	(6.514, 6.568, 6.660)	(5.776, 5.789, 5.901)	(6.360, 6.432, 6.503)

were collected from five runs (one run per day), each lasting four hours. Every ten minutes, the relative measurement errors obtained from these three approaches over the six paths were recorded. The final measurements were produced by averaging the results obtained in the five runs. Again, three metrics, including the mean, median, and MSE of measurement errors were used for performance evaluation, as shown in Table VI. We can observe from Table VI that JitterPath produced smaller relative measurement errors than IGI and PathChirp did.

In summary, we can find that the Internet results were consistent with the simulation results, and that both sets of results indicate that the proposed method (JitterPath) is able to achieve smaller estimation errors.

VI. CONCLUSION

Traditional transport protocols are unable to provide stable transmission due to the lack of information about the available bandwidth. Meanwhile, end-to-end available bandwidth estimation has been found to be helpful for congestion control of multimedia transmission. In this paper, we have proposed a probing noise resilient available bandwidth estimation scheme, called JitterPath, which is based on one-way delay jitter and queuing delay propagation for multimedia applications. To this end, we have presented solutions to 1) queuing region classification; 2) queuing delay propagation; and 3) initial queuing delay determination. We further quantify the captured traffic ratio, which is defined as the ratio of the total output gaps accumulated in JQRs to the total input gaps, and use it to specify the relationship between the probing rate and available bandwidth, i.e., the bottleneck utilization. Finally, a binary search-based probing rate adjustment mechanism has been proposed to iteratively approximate the available bandwidth with an error that is within the preset estimation resolution. Our method has been presented with both the fluid and bursty traffic models. In addition, several simulations and Internet experiments have been conducted to verify our method.

Based on our findings, we have employed the relationship between the probing rate and available bandwidth to determine the cause of packet loss, which is either congestion loss or wireless loss. Our preliminary results are promising. In the future, a TFRC-like rate control mechanism will also be investigated. By integrating both the congestion control and error control technologies, we aim to develop a QoS-guaranteed video transmission system for use in wireless environments.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their comments and suggestions, which have improved the readability of this paper.

REFERENCES

[1] C. Dovrolis, P. Ramanathan, and D. Moore, "What do packet dispersion techniques measure?," in *Proc. Int. Conf. Computer Communication*, Apr. 2001, pp. 905–914.

[2] D. Dutta and Y. Zhang, "An Early Bandwidth Notification (EBN) architecture for dynamic bandwidth environment," in *Proc. IEEE Int. Conf. Communications*, 2002.

[3] N. Hu and P. Steenkiste, "Evaluation and characterization of available bandwidth techniques," *IEEE J. Select. Areas Commun.*, vol. 21, no. 6, pp. 879–894, 2003.

[4] N. Hu and P. Steenkiste, "Improving TCP startup performance using active measurements: Algorithm and evaluation," in *Proc. 11th IEEE Int. Conf. on Network Protocols*, Nov. 2003, pp. 107–118.

[5] N. Hu, L. Li, Z. M. Mao, P. Steenkiste, and J. Wang, "A measurement study of Internet bottlenecks," in *Proc. 24th IEEE Infocom*, Mar. 2005, pp. 41–54.

[6] N. Hu, L. Li, Z. M. Mao, P. Steenkiste, and J. Wang, "Locating Internet bottlenecks: Algorithms, measurements, and implications," in *Proc. ACM SIGCOMM Internet Measurement Conf.*, Oct. 2004.

[7] Y. C. Huang, C. S. Lu, and H. K. Wu, "Reliable available bandwidth estimation based on distinguishing queuing regions and resolving false estimations," in *Proc. IEEE Global Telecommunications Conf. (Globecom)*, Dallas, TX, 2004.

[8] Y. C. Huang, C. S. Lu, and H. K. Wu, "Available bandwidth estimation via one-way delay jitter and queuing delay propagation model," in *Proc. IEEE Wireless Communication and Networking Conference*, Las Vegas, NV, 2006.

[9] V. Jacobson, "Congestion avoidance and control," in *Proc. ACM SIGCOMM*, Sep. 1988, pp. 314–329.

[10] M. Jain and C. Dovrolis, "End-to-end available bandwidth: Measurement methodology, dynamic, and relation with TCP throughput," in *Proc. ACM SIGCOMM*, 2002, pp. 295–308.

[11] M. Jain and C. Dovrolis, "Pathload: A measurement tool for end-to-end available bandwidth," in *Proc. Passive and Active Measurements*, 2002.

[12] M. Jain and C. Dovrolis, "Ten fallacies and pitfalls on end-to-end available bandwidth estimation," in *Proc. Internet Measurement Conf.*, Oct. 2004, pp. 272–277.

[13] S. Keshav, "A control-theoretic approach to flow control," in *Proc. ACM SIGCOMM*, 1991, pp. 3–15.

[14] K. Lai and M. Baker, "Nettimer: A tool for measuring bottleneck link bandwidth," in *Proc. USENIX Symp. Internet Technologies and Systems*, Mar. 2001, pp. 123–134.

[15] Z. G. Li, C. Zhu, N. Ling, X. K. Yang, G. N. Feng, S. Wu, and F. Pan, "A unified architecture for real-time video-coding systems," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 6, pp. 472–487, Jun. 2003.

[16] J. Liu, B. Li, and Y. Q. Zhang, "Adaptive video multicast over the Internet," *IEEE Multimedia*, vol. 10, no. 1, pp. 22–31, 2003.

[17] Q. Liu and J. Hwang, "End-to-end available bandwidth estimation and time measurement adjustment for multimedia QoS," in *Proc. IEEE Int. Conf. on Multimedia and Expo*, 2003.

[18] X. Liu, K. Ravindran, B. Liu, and D. Loguinov, "Single-hop asymptotics in available bandwidth estimation: Sample-path analysis," in *Proc. ACM SIGCOMM Internet Measurement Conf.*, Oct. 2004, pp. 300–313.

[19] B. Melander, M. Bjorkman, and P. Gunningberg, "A new end-to-end probing and analysis method for estimating bandwidth bottlenecks," in *Proc. Global Internet Symp.*, 2000, pp. 415–420.

[20] ns2 [Online]. Available: http://www.isi.edu/nsnam/ns

[21] J. Navratil and R. L. Cottrell, "ABwE: A practical approach to available bandwidth estimation," in *Proc. Passive and Active Measurement Workshop*, 2003.

[22] T. Oetiker and D. Rand, Multi Router Traffic Grapher [Online]. Available: http://www.people.ee.ethz.ch/ oetiker/webtools/mrtg/

[23] V. Paxson, "End-to-end routing behavior in the Internet," *IEEE/ACM Trans. Networking*, vol. 5, no. 5, pp. 601–615, 1997.

[24] PlanetLab [Online]. Available: http://www.planet-lab.org/

[25] R. Puri, K. W. Lee, and K. Ramchandran, "An integrated source transcoding and congestion control paradigm for video streaming in the Internet," *IEEE Trans. Multimedia*, vol. 3, no. 1, pp. 18–32, Feb. 2001.

[26] V. J. Ribeiro, M. Coates, R. H. Riedi, S. Sarvotham, and R. G. Baraniuk, "Multifractal cross traffic estimation," in *Proc. ITC Specialist Seminar on IP Traffic Measurement*, 2000.

[27] V. J. Ribeiro, R. H. Riedi, R. G. Baraniuk, J. Navratil, and L. Cottrell, "PathChirp: Efficient available bandwidth estimation for network paths," in *Proc. Passive and Active Measurement Workshop*, 2003.

[28] V. J. Ribeiro, R. H. Riedi, and R. G. Baraniuk, "Locating available bandwidth bottlenecks," in *Proc. 2nd Int. Workshop on Protocols for Fast Long-Distance Networks*, Oct. 2004.

[29] J. W. Roberts, "Traffic theory and the Internet," *IEEE Commun. Mag.*, vol. 1, no. 39, 2001.

[30] J. Strauss, D. Katabi, and F. Kaashoek, "A measurement study of available bandwidth estimation tools," in *Proc. Internet Measurement Workshop*, Oct. 2003, pp. 27–29.

[31] A. Tirumala and J. Ferguson, Iperf [Online]. Available: http://www.dast.nlanr.net/Projects/Iperf

[32] D. Wu, Y. T. Hou, and Y. Q. Zhang, "Transport real-time video over the Internet: Challenges and approaches," *Proc. IEEE*, vol. 88, no. 12, pp. 1855–1875, Dec. 2000.

[33] N. B. Yoma, J. Hood, and C. Busso, "A real-time protocol for the Internet based on the least mean square algorithm," *IEEE Trans. Multimedia*, vol. 6, no. 1, pp. 174–184, Feb. 2004.

[34] Y. Zhang, N. Duffield, V. Paxson, and S. Shenker, "On the constancy of Internet path properties," in *Proc. ACM SIGCOMM Internet Measurement Workshop*, Nov. 2001.

Yu-Chen Huang was born in Taiwan, R.O.C., on January 4, 1974. He received the B.S. degree and Master degree in Computer Science and Information Engineering from National Chung Cheng University, Taiwan, R.O.C., in 1996 and 1998, respectively. He is currently pursuing the Ph.D. degree in computer science and information engineering at National Central University, Chung-Li, Taiwan, R.O.C.

He is currently a Research Assistant in the Institute of Information Science, Academia Sinica, Taipei, Taiwan. His research interests are in network measurement, networking protocols, and active network services.

Chun-Shien Lu (M'99) received the Ph.D. degree in electrical engineering from National Cheng-Kung University, Tainan, Taiwan, R.O.C., in 1998.

From October 1998 to July 2002, he was with the Institute of Information Science, Academia Sinica, Taiwan, as a Postdoctoral Fellow for his military service. From August 2002 to June 2006, he was an Assistant Research Fellow at the same institute. Since July 2006, he has been an Associate Research Fellow. His current research interests mainly focus on various topics (including security, networking, and signal processing) of multimedia and time-frequency analysis of signals. He is the editor of the book *Multimedia Security: Steganography and Digital Watermarking Techniques for Protection of Intellectual Property*. He has two U.S. patents, two R.O.C. patents, and one Canadian patent in digital watermarking.

Dr. Lu organized a special session on multimedia security in the 2nd and 3rd IEEE Pacific-Rim Conference on Multimedia, respectively (2001–2002). He co-organized two special sessions (in the area of media identification and DRM) in the 5th IEEE International Conference on Multimedia and Expo (ICME), 2004. He was a guest co-editor of the *Journal on Applied Signal Processing* special issue on visual sensor networks in 2005. He has received the paper awards many times from the Image Processing and Pattern Recognition society of Taiwan for his work on data hiding. He was a co-recipient of a National Invention and Creation Award in 2004. He is a member of ACM.

Hsiao-Kuang Wu received the B.S. degree in computer science and information engineering from National Taiwan University in 1989. He received the Master and Ph.D. degrees in computer science from the University of California, Los Angeles (UCLA), in 1993 and 1997.

He is an full Professor of computer science and information engineering at National Central University, Chung-Li, Taiwan. His primary research interests include wireless networks, mobile computing, and broadband networks.

Dr. Wu is a member of Institute of Information and Computing Machinery (IICM).