Reliable Available Bandwidth Estimation Based on Distinguishing Queuing Regions and Resolving False Estimations

Yu-Chen Huang[1,2], Chun-Shien Lu[1], Hsiao-Kuang Wu[2]
[1]Institute of Information Science, Academia Sinica, Taipei, Taiwan, ROC
[2]Dept. Computer Sci. & Information Eng., National Central Uni., Chung-Li, Taiwan, ROC

Abstract-Video transmission needs stable sending rate in order that video could be displayed to show uniform qualities. In addition, lower packet loss rate is rather helpful in reducing video quality degradation. Therefore, reliable available bandwidth estimation becomes an indispensable step towards robust transmission of multimedia data. The existing available bandwidth estimation methods cannot deal with the false estimation problem and thereby precise estimation is impossible. In this paper, we propose a reliable available bandwidth estimation method based on distinguishing queuing regions and resolving false estimations. Promising simulation results indicate that our method can obtain available bandwidth precisely no matter the network environment is single-bottleneck or multiple-bottleneck.

I. Introduction

Bandwidth is one of precious resources in the network. In order to exploit network bandwidth efficiently, multimedia data will be compressed in advance before it is transmitted over the network. However, when packet loss occurred, the quality of decoded data will be degraded. In order to display video with uniform qualities, the network needs to provide Quality-of-Service (QoS) [18]. In this paper, we consider QoS at the application level.

Numerous approaches were proposed to approximately make use of network bandwidth. In [15], Dutta and Zhang proposed to improve the current Internet infrastructure and to support different kinds of QoS in the core network. However, their method undesirably changes the network intermediate nodes leading to hard implementation problem. Other approaches [16]~[19] were proposed to adjust the transmission rate according to the current network conditions. Their common characteristic is that the scalable coding scheme associated with network behavior monitor was adopted to detect network conditions and change data rate accordingly. The key is how to correctly detect network conditions (among which congestion plays a crucial role). When network is congested (which means the traffic arrival rate is larger than the departure rate), the incoming packets will be queued in the routers. Once the routers overflow, packets will be lost and the senders will need to wait a long packet timeout to retransmit the lost packets. Excessive delay makes the receiver drop those delayed packets to create packet loss. In order to deal with the congestion problem, traditional transport protocols (e.g., TCP and UDP) are still insufficient for video transmission because they ignore the network conditions so that problems remain when packet loss appears.

In order to reduce congestion, we prefer to handle this problem from controlling sending rate by means of available bandwidth estimation. Our viewpoint is that bandwidth utilization and variations are actually related with sending rate, congestion, and packet loss. This viewpoint is also supported in [5][6]. Keshav [3] first brought a new idea, called "packet-pair'', to estimate the bottleneck bandwidth, which is defined as the minimum bandwidth among the links from the sender to the receiver. However, bottleneck link is not really crucial enough to affect network conditions. On the contrary, available bandwidth has been recognized to be the most important factor. In [2], the available bandwidth is determined as the minimum bandwidth that has not used. Thus, it is known that the bottleneck link may be not the link with minimum available bandwidth.

In order to estimate the available bandwidth, some approaches were presented to actively send probe packet trains to interact with the competing traffic and to look for the turning point by analyzing the relation between competing traffic and probe packet train. The "turning point" specifies the "rate" where the network service rate is equal to the departure rate. The existing active bandwidth estimation tools can be divided into two types [4]: (1) probe gap model (PGM) and (2) probe rate model (PRM).

In PGM methods [4],[8]~[10], a packet pair was sent into the network in the hope that it can be expanded with the competing traffic. If the size of the competing traffic placed in between the packet pair could be known, it is possible to estimate the available bandwidth as the bottleneck bandwidth subtracting the competing traffic throughput. PGM makes two assumptions in order to guarantee reliable available bandwidth estimation. The two assumptions include (1) the link with the minimum available bandwidth is the same as the bottleneck link; (2) the packet pair must be closely placed with the competing traffic. Based on these two assumptions, PGM can estimate available bandwidth quickly without pouring large probe traffics into the network. However, the first assumption does not hold in a multiple-bottleneck environment where the link with the minimum available bandwidth may have been changed but the receiver does not know this variation.

On the other hand, PRM methods [7][11][13] exploited self-induced congestion to detect available bandwidth. Specifically, when the traffic probing rate is larger than the available bandwidth, the queue at the bottleneck link begins to grow such that probe packet is forced to be delayed. In addition, the probe packet begins to increase its delay once the probe traffic rate is over the turning point. At this moment, the available bandwidth is defined to be the transmission rate of probe traffic at the turning point. However, PRM needs to pour more packet pairs to obtain reliable estimation such that it incurs intrusiveness and needs long convergence time. On the other hand, PRM doesn't rely on the first assumption required for PGM such that PRM is suitable for both the single-bottleneck and multiple-bottleneck environments.

0-7803-8794-5/04/$20.00 (C) 2004 IEEE

However, both PGM and PRM lead to the "false estimation" problem. Fig. 1 shows this problem for PRM. To be precise, it is possible for both the probing and competing traffics to enter into the bottleneck in an overlapping manner in time to give rise to the one-way delay (OWD) increasing trend in the probe traffic. Under this situation, even the sum of both the competing and probing traffics is not larger than the bottleneck bandwidth the available bandwidth will be falsely estimated due to the detected OWD increasing trend.

Some selected PGM- and PRM-based available bandwidth estimation methods are compared and depicted in Table 1. From this table, it is obvious that the common false estimation problem is ignored and un-solved in the literature. In view of this, the aim of this paper is to propose a reliable available bandwidth scheme that takes the false estimation problem into consideration. We provide a new policy to distinguish different queuing regions and define the so-called "queuing region factor (*QRF*)" to analyze the relationship between probe packets in order to eliminate "false estimation." Besides, our idea is ready to be merged with the existing (either PGM or PRM) methods to further improve their inherent disadvantages, as shown in Table 1.

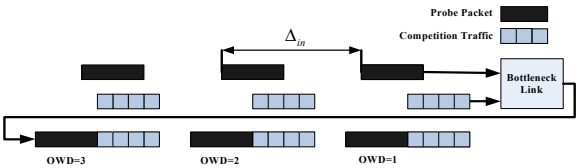

Fig. 1. Practically, the probe rate is smaller than the available bandwidth. However, one-way delay increasing trend is detected to wrongly claim that the probe rate is larger than the available bandwidth. This "false estimation" problem is found in PRM.

II. Proposed Method

In this section, our ideas of distinguishing queuing regions and resolving false estimation will be described in PRM. However, we would like to emphasize that our method can also be merged with PGM.

II. 1 One-Way Delay Increasing Trend

When the sender sends probe packets to the receiver, the time (one-way delay, OWD), D_k, that is needed to propagate probe packet k with size L_k is expressed as:

$$D_k = \sum_{i=1}^{N}\left(\frac{L_k}{C_i} + d_i^k + \sigma_i\right), \tag{2}$$

where C_i is the bandwidth of link i, d_i^k denotes the queuing delay of probe packet k at router i, σ_i denotes processing delay at router i, and N is the number of links from the sender to the receiver. In Eq. (2), both $\sum_{i=1}^{N}\frac{L_k}{C_i}$ and $\sum_{i=1}^{N}\sigma_i$ are constants for the same probe packet propagated along the same path. As a result, we can derive that the total queuing delay of the probe packet k, $\sum_{i=1}^{N} d_i^k$, fully depends on its propagation time, D_k.

Conventionally, one-way delay increasing trend (DI_{trend}) appears if the following propagation time relationship exists

$$\begin{aligned} &DI_{trend} > 0, \ if \ P\{D_l > D_m\} > 0.5; \\ &DI_{trend} = 0, \ if \ P\{D_l > D_m\} = 0.5; \quad \forall l > m \\ &DI_{trend} < 0, \ if \ P\{D_l > D_m\} < 0.5, \end{aligned} \tag{3}$$

where *l* and *m* denote the indices of probing packets in the packet train. Based on Eq. (3), the relationship between available bandwidth (*avbw*) and probe rate ($probe_{rate}$) could be built by means of the following rules:

$$\begin{aligned} &probe_{rate} < avbw, \quad \text{if } DI_{trend} < 0; \\ &probe_{rate} > avbw, \quad \text{if } DI_{trend} > 0; \\ &probe_{rate} \approx avbw, \quad \text{if } DI_{trend} = 0. \end{aligned} \tag{4}$$

Table I. Comparisons between Selected Available Bandwidth Estimation Methods

	Probe Gap Model				**Probe Rate Model**			
	Spruce [4]	**AbwE [9]**	**IGI [8]**	**Delphi [10]**	**Pathload [11]**	**pathChirp [12]**	**Pathbw [7]**	**Proposed Method**
Intrusiveness	low	low	low	low	high	low	high	middle
Single Bottleneck Assumption	yes	yes	yes	yes	no	no	no	no
Estimation Resolution	high	high	high	high	low	low	high	high
Handle False Estimation	*no*	*no*	*no*	*no*	*no*	*no*	*no*	*yes*

II. 2 Queuing Region (QR) Determination

The major contribution of proposed method is to determine the types of a queuing region between two probe packets. The "queuing region" is defined as the time duration between two probe packets during which the queue is either empty or not. If a queuing region is empty, it is called a "disjoint queuing region (DQR);" otherwise, it is a "joint queuing region (JQR)". We have observed from probe packets that the accumulated queuing delays are helpful in judging the queuing region in each probe packet gap. In order to better analyze the relationship between probe rate and queuing region, our analyses will be conducted in a single-bottleneck environment. However, we would like to emphasize that our method still works in a multiple-bottleneck environment.

II. 2. 1 Queuing Region Factor

In similar to Eq. (4), we will define a relationship between probe rate and available bandwidth by exploiting the queuing region factor (QRF). First, we calculate the number of DQR and JQR, $\#QR_{DQR}$ and $\#QR_{JQR}$, respectively, from the probe packet train at the receiver side. Then, the so-called queuing region factor is defined according to queuing theory as:

$$QRF = \frac{\#QR_{JQR}}{\#QR_{JQR} + \#QR_{DQR}}, \qquad (5)$$

where $0 \le QRF \le 1$. QRF can be used to specify the relationship between available bandwidth and probe rate as:

$$\begin{aligned} probe_{rate} &< avbw, \text{ if } QRF < 0.5; \\ probe_{rate} &> avbw, \text{if } QRF > 0.5; \\ probe_{rate} &\approx avbw, \text{if } QRF = 0.5. \end{aligned} \qquad (6)$$

Although Eq. (4) is an equivalent of Eq. (6), Eq. (6) is even extremely useful in dealing with the false estimation problem, as described in Sec. I. How to determine the types of queuing regions will be investigated in the next subsection.

II. 2. 2 Policies for Classifying Queuing Region

The policies used to classify queuing regions will be described in the subsection. There will be four policies designed according to the relationship between the initial packet gap (Δ_{in}), the obtained packet gap (Δ^i) exactly after bottleneck, and the bottleneck gap (Δ_B). When the transmission path is chosen, the bottleneck router will be kept un-changed until the route is changed again. Therefore, the bottleneck bandwidth only needs to be updated when a route is changed. Some tools like NetDyn probes [20], packet pairs [21], bprobe [22], and nettimer [23] can be used to retrieve the bottleneck bandwidth.

II. 2. 2. 1 Initial packet gap smaller than obtained packet gap after bottleneck, $\Delta_{in} < \Delta^i$.

Policy 1 ($\Delta_{in} < \Delta^i$):

The first policy will be used when the gap of a probe packet pair is totally filled/expanded with the competing traffic, as shown in Fig. 2, where Δ^i denotes the gap between probe packets i and $i-1$. In this case, Δ^i is larger than Δ_{in} and the duration of Δ^i forms a JQR. We can infer from this case that the gap Δ^i is filled with the competing traffic and the probe traffic rate ($probe_{rate}$) is larger than the available bandwidth (*avbw*). Therefore, the probe packet i will be queued in a router.

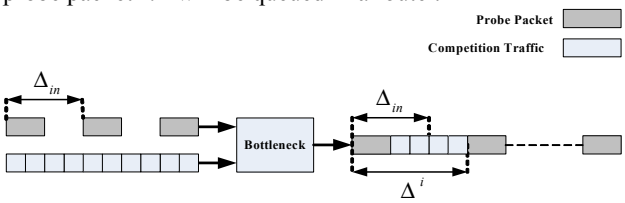

Fig. 2. Competition traffic expands the gap Δ^i and Δ^i is larger than Δ_{in}.

II. 2. 2. 2 Initial packet gap larger than or equal to obtained packet gap after bottleneck, $\Delta^i \le \Delta_{in}$.

There will be three different cases for the initial packet gap that is larger than or equal to the packet gap obtained after the packet passes the bottleneck. They are, respectively, discussed as follows.

Policy 2 ($\Delta_B = \Delta^i < \Delta_{in}$):

Fig. 3 depicts the case that $\Delta_B = \Delta^i < \Delta_{in}$, which means that the previous queuing delay shortens the gap between probe packets $i-1$ and i, and the duration of Δ^i is a JQR.

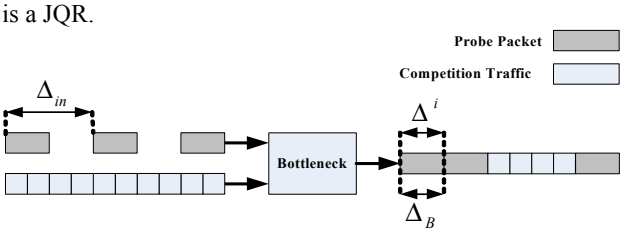

Fig. 3. Competition traffic arrives before the probe packet $i-1$ and shortens the gap Δ^i to a minimum value, which is equal to Δ_B (which denotes the transmission delay of the probe packet at the bottleneck output link**).**

Policy 3 ($\Delta_B < \Delta^i = \Delta_{in}$):

When the initial packet gap is equal to the packet gap obtained after the packet passes the bottleneck, as shown in Fig. 4, we cannot infer what kind of a queuing region Δ^i implies from Δ^i itself. The reason is that we never know whether Δ^i is empty or filled with the competing traffic. Under this condition, we need the information coming from Δ^{i-1} to assist our decision. If Δ^{i-1} is larger than Δ_{in}, we can infer that Δ^{i-1} is filled with the competing traffic and the additional queuing delay will propagate to Δ^i. Hence, Δ^i should also be filled with the competition traffic to be determined as JQR. If no helpful information could be obtained from Δ^{i-1}, we need to utilize the already known subsequent gap Δ^{i+1}. At this moment, Δ^i is initially assumed to be a DQR. If Δ^{i+1} is larger than Δ_{in}, then the duration of Δ^{i+1} is a JQR and it will compress Δ^i to

make Δ^i a JQR. If Δ^{i+1} is still un-decided, we refer to Δ^{i+2}. According to our observations and saving of convergence time, it is enough for us to determine the type of queuing regions for Δ^i by visiting until Δ^{i+2}.

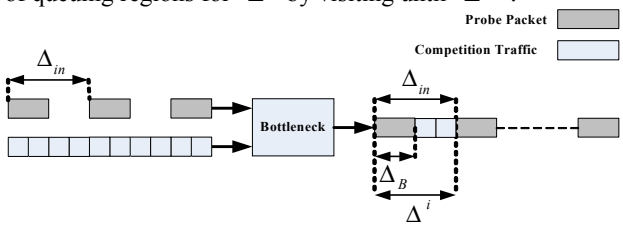

Fig. 4. The packet gap Δ^i is equal to initial gap Δ_{in} and Δ^{i-1} is larger than Δ_{in}.

Policy 4 ($\Delta_B < \Delta^i < \Delta_{in}$):

The last case is $\Delta_B < \Delta^i < \Delta_{in}$, as shown in Fig. 5. Under this situation, we are confident to conclude that there is a competing traffic to delay the probe packet *i-1* and thereby the gap Δ^i is compressed. However, we cannot determine whether Δ^i is filled with the competing traffic or not. Therefore, we must refer to either Δ^{i-1} or Δ^{i+1} to determine the queuing region type for Δ^i. Due to the limit of space, we briefly summarize the rules as follows:

$$\Delta^i \text{ is } \begin{cases} NQR, \text{ if } \Delta^{i-1} > \Delta_{in} \text{ and } \Delta^i = 2\Delta_{in} - \Delta^{i-1}; \\ NQR, \text{if } \Delta^{i-1} > \Delta_{in} \text{ and } \Delta^i < 2\Delta_{in} - \Delta^{i-1}; \\ JQR, \text{if } \Delta^{i-1} > \Delta_{in} \text{ and } \Delta^i > 2\Delta_{in} - \Delta^{i-1}; \\ JQR, \text{if } \Delta^{i-1} = \Delta_{in}; \\ NQR, \text{if } \Delta_B < \Delta^{i-1} < \Delta_{in} \text{ and } \Delta^i < \Delta^{i-1}; \\ JQR, \text{if } \Delta_B < \Delta^{i-1} < \Delta_{in} \text{ and } \Delta^i > \Delta^{i-1}; \\ NQR, \text{if } \Delta_B < \Delta^{i-1} < \Delta_{in} \text{ and } \Delta^i = \Delta^{i-1}; \\ JQR, \text{if } \Delta^{i-1} = \Delta_B; \end{cases} \quad (7)$$

where NQR denotes that Δ^i has to be determined from its subsequent Δ^{i+1}.

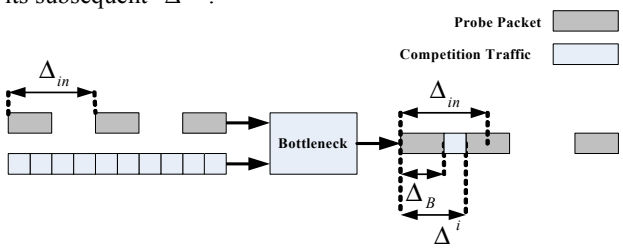

Fig. 5. The gap Δ^i is smaller than the initial gap Δ_{in} and is larger than the bottleneck gap Δ_B.

II.3 Probe Packet Rate Adjustment

When the receiver receives a probe packet train, he/she may exploit wither the one-way delay increasing trend (Eq. (4)) or the queuing region factor (Eq. (6)) for probe rate adjustment. However, the contribution of the proposed *QRF* is that *QRF* can help DI_{trend} in detecting two false estimations, *i.e.,* false positive (the fact that probe rate is larger than *avbw* is wrongly determined as that probe rate is not larger than *avbw*) and false negative (the fact that probe rate is smaller than *avbw* is wrongly determined as that probe rate is not smaller than *avbw*). Our simulation results (discussed in Sec. IV) further confirm the power of *QRF*. Here, false positive and false negative are defined by referring to Eqs. (4) and (6) as

$$\begin{aligned} &\textit{false positive}, \ \textit{if } DI_{trend} < 0 \ \& \ QRF > 0.5; \\ &\textit{false negative}, \ \textit{if } DI_{trend} > 0 \ \& \ QRF < 0.5. \end{aligned} \quad (8)$$

The capability of *QRF* in detecting false estimations shows its advantage over the OWD increasing trend.

We will use *QRF* (Eq. (6)) to determine the relationship between probe rate and available bandwidth. Then, the next probe rate at time *t+1* could be adjusted according to the following binary search rule:

$$\begin{aligned} &probe_{rate}^{\min} = probe_{rate}(t) \text{ if } probe_{rate}(t) > avbw; \\ &probe_{rate}^{\max} = probe_{rate}(t) \text{ if } probe_{rate}(t) \le avbw; \\ &probe_{rate}(t+1) = (probe_{rate}^{\min} + probe_{rate}^{\max})/2; \end{aligned} \quad (9)$$

where $probe_{rate}^{\min}$ is the minimum probe traffic rate that is detected when OWD increasing trend is present, and $probe_{rate}^{\max}$ is the maximum probe traffic rate that is detected when OWD increasing trend is absent. In Eq. (9), the initial values of $probe_{rate}^{\max}$ and $probe_{rate}^{\min}$ are set to *0* and bottleneck bandwidth, respectively. The above probe rate adjustment procedure stops if $| probe_{rate}^{\min} - probe_{rate}^{\max} | < \varepsilon$ holds, where ε is a small number used to define how the probe rate could be adjusted to approximate the true available bandwidth.

II. 4 How to Handle Packet Loss?

The proposed method tries to look for the turning point by pouring traffic to the network within a short time. Once the buffer of a router overflows, the incoming data will be dropped to create packet loss. Here, it is assumed that packet loss is caused by congestion. Under this circumstance, the probe rate is adjusted based on Eq. (9) but the values of $probe_{rate}^{\max}$ and $probe_{rate}^{\min}$ are kept un-changed in order to avoid random packet loss.

II. 5 Is the Proposed Method intrusive?

One of the most important factors that affects network behaviors is the injected probing traffics. Since TCP uses AIMD to adjust its sending rate by monitoring packet loss, if the rate of injected probing traffics is larger than the available bandwidth, then the un-delivered traffics will be queued and then dropped when the queue overflows. Under this circumstance, TCP will reduce its sending rate and the network behavior is changed due to the injected probing traffics. The size of injected probing traffics is determined by the length of the packet train. We have observed that the size of a packet train play s trade-off between intrusiveness and accuracy of available bandwidth. This problem has been investigated in our extended work [24].

III. Simulation Results

Our simulations were conducted using ns2 [14]. The proposed method was also compared with pathChirp [12] in term of accuracy of available bandwidth estimation. pathChirp is selected because it, strictly speaking, is a hybrid of PGM and PRM to possess both advantages. Our method and pathChirp were evaluated with respect to two

configurations (Fig. 6 and Fig. 9) and two different competing traffics (constant bit-rate (CBR) traffic and FTP traffic). The packet size of the CBR traffic is fixed as *1*Mbps. The FTP traffic, a kind of TCP flows, guarantees reliable transmission with rate adjustment through additive increasing and multiplicative decreasing (AIMD) [1]. Hence, the rate of FTP traffic will be stable when the available bandwidth is unchanged [5].

III. 1 Single-Bottleneck Network Model

The first topology was set in a single bottleneck environment, as shown in Fig. 6, where Ps and Pr, respectively, denote the probe traffic sender and receiver. In Fig. 6, the bottleneck bandwidth is set to be *10*Mbps. The competing traffic flows from Cs to Cr. At the first *50* seconds, the competing traffic's rate is kept *0.5*Mbps and changed to *1*Mbps from 51^{st} to 100^{th} seconds. After that, the competing traffic's rate increases *1*Mbps per *50* seconds until the end of 500^{th} second.

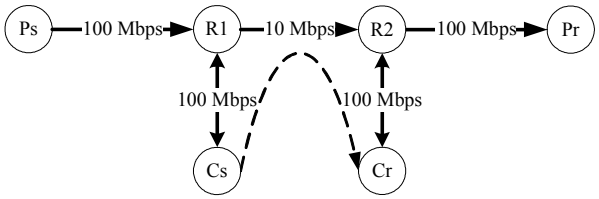

Fig. 6. Single-bottleneck network model. Bottleneck link is equal to the link with the minimum available bandwidth. Ps and Pr denote probe traffic sender and receiver, respectively. The competition traffic, flowing from Cs to Cr, is with a constant-bit-rate (CBR).

In Fig. 7, the results of estimated available bandwidth with respect to the proposed method and pathChirp, and the true *avbw* are plotted for comparisons under the CBR traffic only. It can be observed that our method is able to approximate the actual available bandwidth accurately. However, pathChirp gives over-estimations when the traffic load is low and gives improved but oscillated results when network traffic is high. The main reason is that pathChirp wrongly converges when false estimations (Eq. (8)) occur. In Fig. 8, we show the estimation results under the situation that the competing traffic was composed of the CBR and FTP traffics. The results of the estimated available bandwidth obtained from our method and pathChirp, and the true *avbw* are plotted for comparisons. It can be observed from Fig. 8 that our method is able to approximate the true available bandwidth stably and doesn't yield over-estimations. However, pathChirp generates over-estimations when the competing traffic's load is low and gives improved but oscillated estimations when the competing traffic's load is high. Again, this is due to that pathChirp wrongly converges when false estimations occur.

III. 2 Multiple-bottlenecks Network Model

In the second network topology, we focus on the multiple-bottleneck network environment, as shown in Fig. 9. The multiple-bottleneck model is used to manifest the advantage of PRM-like methods. In Fig. 9, the bottleneck bandwidth is *10*Mbps (R1~R2) and the bandwidth of the link (R2~R3) is *20*Mbps. The link with the minimum available bandwidth is from R1 to R2. The throughput of the competing traffic sent from C1s to C1r is *3*Mbps.

In addition, the initial rate of the competing traffic from C2s to C2r is *5*Mbps and is then is adjusted with an increasing rate *1*Mbps per *50* seconds. It should be noted that the link with the minimum available bandwidth will be shifted from link R1~R2 to the link R2~R3 at the 250^{th} second in this multiple-bottleneck model.

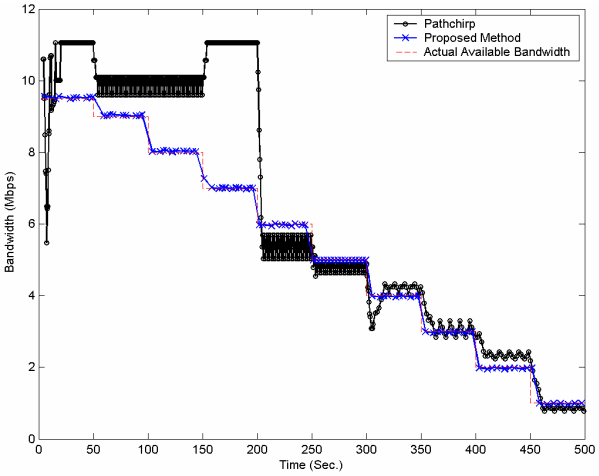

Fig. 7. Comparison of available bandwidth estimation: proposed method vs. pathChirp on the single-bottleneck network model (Fig. 6).

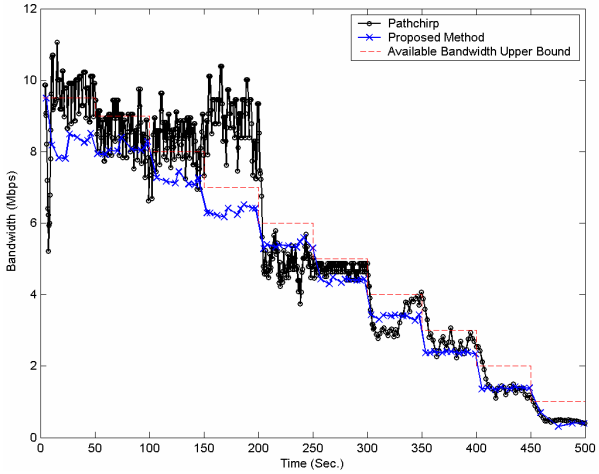

Fig. 8. Comparison of available bandwidth estimation: proposed method vs. pathChirp on the single-bottleneck network model (Fig. 6). The competing traffics include the CBR traffic and FTP traffic.

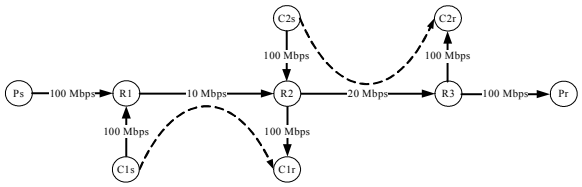

Fig. 9. Multiple-bottlenecks network model. Bottleneck link is not equal to the link with the minimum available bandwidth. Both cross traffics are constant-bit-rate (CBR).

In Fig. 10, the results of estimated available bandwidth with respect to the proposed method and pathChirp, and the true *avbw* are plotted for comparisons. It can be observed that our method is able to approximate the actual available bandwidth quite accurate. However, pathChirp gives under-estimations when the traffic load is low and gives improved but oscillated results when network traffic is high. Again, pathChirp cannot detect false estimations in a multiple-bottleneck model environment. In Fig. 11, we show the estimation results under the situation that the competing traffic was composed of the CBR and FTP traffics. The results of the available bandwidth estimated from our method and pathChirp, and the true *avbw* are

plotted for comparisons. It can be observed from Fig. 11 that both methods basically yield under-estimations. However, compared with pathChirp our result is closer to the actual available bandwidth.

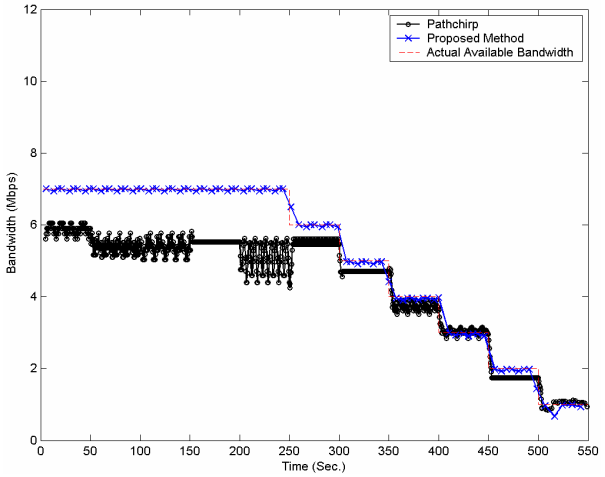

Fig. 10. Comparison of available bandwidth estimation: proposed method vs. pathChirp on the multiple-bottleneck network model (Fig. 9). The competing traffic contains the CBR traffic only.

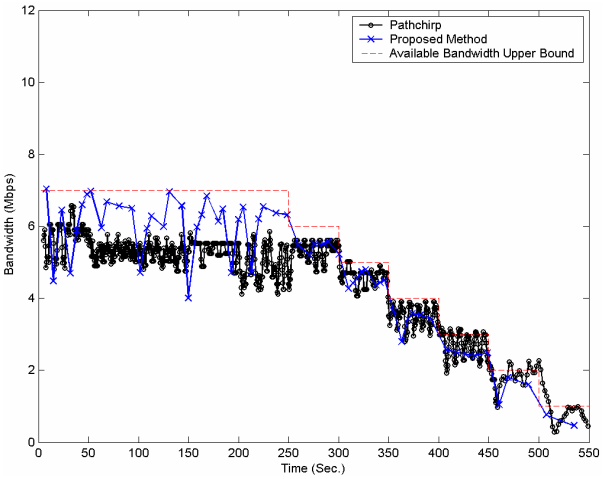

Fig. 11. Comparison of available bandwidth estimation: proposed method vs. pathChirp on the multiple-bottleneck network model (Fig. 9). The competing traffic includes both the CBR and FTP traffics.

IV Concluding Remarks

Traditional transport protocols are unable to provide stable video transmission. The main reason is that the information about available bandwidth is ignored. However, end-to-end available bandwidth estimation plays an important role in helping congestion control of multimedia transmission. In this paper, we have proposed a reliable available bandwidth estimation method based on (1) distinguishing queuing regions; and (2) resolving false estimations. Since the false estimations, resulted from OWD increasing trend, in specifying the relationship between probe rate and available bandwidth could be corrected by our method, the available bandwidth can be measured accurately. This characteristic explains the major difference between our method and the existing methods. Furthermore, our method can make the probe gap model-based methods work in the multiple-bottleneck model. More analyses of available bandwidth estimation can be found in [24].

References

[1] V. Jacobson, "Congestion Avoidance and Control," *Proceedings of ACM SIGCOMM*, pp.314-329, 1988.

[2] M. Jain and C. Dovrolis, "End-to-End Available Bandwidth: Measurement Methodology, Dynamic, and Relation with TCP Throughput," *Proceedings of ACM SIGCOMM*, 2002.

[3] S. Keshav, "A Control-Theoretic Approach to Flow Control," *ACM SIGCOMM*, pp. 3-15, September 1991.

[4] J. Strauss, D. Katabi, and F. Kaashoek, "A Measurement Study of Available Bandwidth Estimation Tools," *Proc. Internet measurement Workshop*, pp. 27-29, 2003.

[5] Y. Zhang, N. Duffield, V. Paxson, and S. Shenker, "On the Constancy of Internet Path Properties," Proceedings of *ACM SIGCOMM Internet Measurement Workshop*, Nov. 2001

[6] V. Paxson, "End-to-end Routing Behavior in the Internet," *IEEE/ACM Transactions on Networking*, Vol. 5, 1997.

[7] Q. Liu, and J. Hwang, "End-to-End Available Bandwidth Estimation and Time Measurement Adjustment for Multimedia QoS," *Proc. IEEE Int. Conf. on Multimedia and Expo*, 2003.

[8] N. Hu and P. Steenkiste, "Evaluation and Characterization of Available Bandwidth Techniques," *IEEE Journal on Selected Areas in Communication: Special Issue in Internet and WWW Measurement, Mapping, and Modeling*, 2003.

[9] J. Navratil, and R. L. Cottrell, "ABwE: A Practical Approach to Available Bandwidth Estimation," *Proc. Passive and Active Measurement Workshop*, 2003.

[10] V. J. Ribeiro, M. Coates, R. H. Riedi, S. Sarvotham, and R. G. Baraniuk, "Multifractal Cross Traffic Estimation," *Proc. of ITC specialist seminar on IP traffic Measurement*, 2000.

[11] M. Jain and C. Dovrolis, "Pathload: A Measurement Tool for End-to-End Available Bandwidth," *Proc. Passive and Active Measurements*, March 2002.

[12] V. J. Ribeiro, R. H. Riedi, R. G. Baraniuk, J. Navratil, and L. Cottrell, "pathChirp: Efficient Available Bandwidth Estimation for Network Paths," *Proc. Passive and Active Measurement Workshop*, 2003.

[13] B. Melander, M. Bjorkman, and P. Gunningberg, "A New End-to-End Probing and Analysis Method for Estimating Bandwidth Bottlenecks," *Proc. Global Internet Symposium*, 2000.

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

[15] D. Dutta, Y. Zhang, "An Early Bandwidth Notification (EBN) Architecture for Dynamic Bandwidth Environment," *Proc. IEEE Int. Conf. on Communications*, 2002.

[16] 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. on Circuits and Systems for Video Technology*, 2003.

[17] R. Puri, K. W. Lee, and K. Ramchandran, "An Integraded Source Transcoding and Congestion Control Paradigm for Video Streaming in the Interner," *IEEE Trans. on Multimedia*, 2001.

[18] D. Wu, Y. T. Hou, and Y. Q. Zhang, "Transport Real-Time Video over the Internet: Challenges and Approaches," *Proceedings of the IEEE*, 2000.

[19] J. Liu, B. Li, and Y. Q. Zhang, "Adaptive Video Multicast over the Internet," *IEEE Multimedia,* Vol. 10, 2003.

[20] J.-C. Bolot, "End-to-end packet delay and loss behavior in the Internet," in *Proc. ACM SIGCOMM Symp. Communications Architectures Protocols*, pp. 289–298, 1993.

[21] S. Keshav. Packet Pair Flow Control. [Online]. Available:http://www.cs.cornell.edu/skeshav/doc/94/2–17.ps

[22] R. L. Carter and M. E. Crovella, "Measuring bottleneck link speed in packet-switched networks," Boston Univ., Boston, MA, Comput. Sci. Dept., Tech. Rep., Mar. 1996.

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

[24] Y. C. Huang, C. S. Lu, and H. K. Wu, ``Queuing Delay Propagation Model (QDPM)-based Available Bandwidth Estimation for Multimedia QoS,'' submitted to *Infocom* 2005.