

Fast End-to-End Available Bandwidth Estimation for Real-Time Multimedia Networking

Sheng-Shuen Wang

Department of Computer Science
National Chiao Tung University
Hsinchu, Taiwan
sswang@csie.nctu.edu.tw

Hsu-Feng Hsiao

Department of Computer Science
National Chiao Tung University
Hsinchu, Taiwan
hillhsiao@cs.nctu.edu.tw

Abstract—Dynamic bandwidth estimation serves as an important basis for performance optimization of real-time distributed multimedia applications. The objective of this paper is to develop a bandwidth estimation algorithm for the fast fluctuated Internet. We analyze the relationship between the one way delay and the dispersion of packets train, and propose an available-bandwidth estimation algorithm which makes use of these two features without requiring administrative access to the intermediate routers along the network path. Instead of binary search or fixed-rate bandwidth adjustment of the probing data as described in literature, we use *top-down* approach to infer available bandwidth robustly and much more efficiently.

Keywords—packet train; bandwidth estimation; available bandwidth; one way delay; packet dispersion

Topic area—multimedia networking

I. INTRODUCTION

With the rapid growth of the Internet, bandwidth estimation plays an important role for the growing demand of multimedia services, such as video conference, streaming video on demand, interactive IPTV, VoIP, etc. The knowledge of available bandwidth of the bottleneck link is crucial in terms of improving Quality of Service (QoS) in many distributed applications, such as the overlay construction of peer to peer system, optimization of dynamic server selection, socket buffer sizing, admission control, and congestion control for the streaming applications. One can utilize Multi-Router Traffic Grapher (MRTG) to estimate the available bandwidth from intermediate routers. However, it is often difficult if possible due to various technical and privacy considerations or due to an insufficient level of measurement resolution.

Among the emerging research in bandwidth estimation, link capacity and available bandwidth are of interest. The prior is constrained by the underlying transmission bandwidth physically. Given that packets are delivered from sender S to its receiver R through a fixed and unique network path P , which consists of a sequence of first-come-first-served and store-and-forward links, the narrow link of a network path P is defined as the link with minimum capacity along the path. The capacity C of the narrow link is:

$$C = \min_{i=1 \dots H} C_i, \quad (1)$$

where C_i is the link capacity of link i , and H is the number of hops in P .

On the other hand, available bandwidth depends on the traffic load of the path and it is typically a time-varying random variable. Assume $\lambda_i(t)$ is the traffic load of link i at time t , the available bandwidth $A_i(t, T)$ of link i is the average unused bandwidth over some time interval T as shown in (2).

$$A_i(t, T) = \frac{1}{T} \int_t^{t+T} (C_i - \lambda_i(t)) dt. \quad (2)$$

The available bandwidth $A(t, T)$ of the tight link, which is defined as the link with minimum available bandwidth along a path, is:

$$A(t, T) = \min_{i=1 \dots H} A_i(t, T). \quad (3)$$

For multimedia streaming over Internet, audio/speech stream is usually more important than video stream and its bit rate, which remains constant such as the speech stream encoded by G.723.1, is much less. Thus, we can use audio packets as packet train [1] to estimate available bandwidth by manipulating the packet dispersion accordingly at the transmitter as proposed in the latter sections. When the available bandwidth is inferred dynamically, it can be used by rate-based distributed multimedia applications, such as streaming service of scalable video coded contents, for congestion control purposes.

The rest of this paper is organized as follows. In Section II, we briefly review well-known bandwidth measurement algorithms in literature and analyze important features to infer available bandwidth. We present the proposed algorithm in Section III, followed by the simulation results and concluding remarks in Section IV and Section V, respectively.

II. RELATED WORK AND ANALYSIS

Several studies have been devoted to the research of available bandwidth estimation in recent years. Except for the network mathematic model which is based on some specified network behavior or protocol [2], probing-based methods by means of packet train [1] analysis are widely adopted to infer network utilization. Packet train is a sequence of probing packets of equal packet size and the probing packets are arranged either back-to-back or with some specified inter-packet dispersion. According to the analysis domains, there

are two major types of packet-train based algorithms: one-way-delay (OWD) based analysis model and dispersion based analysis model.

A. One-Way-Delay Based Analysis Model:

Given that a sender S transmits K packets of packet size L to its receiver R , the OWD D^k of k -th packet can be modeled as the summation the transmission delay (L/C_i), processing delay (σ_i), and queuing delay (d_i^k) of each and every link ($i=1 \dots H$) along the path.

$$D^k = \sum_{i=1}^H \left(\frac{L}{C_i} + \sigma_i + d_i^k \right). \quad (4)$$

The OWD difference between adjacent packets can be expressed as the contribution from queuing delay as shown in (5).

$$\Delta D^k = D^{k+1} - D^k = \sum_{i=1}^H (\Delta d_i^k). \quad (5)$$

The idea of OWD based model is based on the following proposition [7]:

$$\text{If } R_p > A, \Delta D^k > 0,$$

$$\text{If } R_p \leq A, \Delta D^k = 0.$$

where R_p stands for the probing rate and A is the available bandwidth of a given path. The proposition concludes that if the probing rate is slower than the available bandwidth of the path, the arrival rate at the receiver will match their probing rate at the sender. On the other hand, if the probing rate is faster than the available bandwidth, then network queues will build up and the probing packets will be delayed ($\Delta D^k > 0$). By observing the delay trend of OWD, many algorithms, e.g. Pathload[3], pathChirp[4], Pathbw[5], TOPP[6] and SLoPS[7], search for the turning point at which the sending rate and receiving rate start to match.

B. Dispersion Based Analysis Model:

Dispersion based model exploits the information of the inter-arrival time between two successive probing packets at the receiver. Let δ_{in} and δ_{out} be the time dispersion of a packet pair before and after passing through a single hop, respectively. Assume that the network queue will not be empty between the departure time of the first probing packet of a packet pair and the arrival time of the second probing packet in the joint queuing region (JQR) [8]. Given the network capacity of the tight link C , the available bandwidth $A (=C-\lambda)$ can be estimated by solving the following equation for the traffic load λ [9].

$$\delta_{out} = \frac{L}{C} + \frac{\lambda}{C} \delta_{in}. \quad (6)$$

However, if these two packets do not fall into the same period, (i.e., in the disjoint queuing region (DQR) [8]), the packet dispersion before and after passing through a hop will be equal ($\delta_{out} = \delta_{in}$). We denote $R_i = L/\delta_i$ as the departure rate after passing through hop i with packet time dispersion δ_i . For the probing packets passing through hop i with the arrival rate

R_{i-1} to hop i and departure rate R_i from the same hop, R_{i-1} and R_i will have the following relationship,

$$R_i = R_{i-1} \frac{C_i}{\lambda_i + \max\{R_{i-1}, A_i\}} \quad (7)$$

where λ_i is the traffic load of hop i .

Obviously, the departure rate will be less than or equal to the arrival rate ($R_{i-1} \geq R_i$).

Pathrate[1] has proved that if $R_{i-1} \geq A_i$, then $R_i \geq A_i$. Because $R_{i-1} \geq A_i$ and $C_i \geq A_i$, the inequality equation in (8) holds. It also implies that $R_i \geq A_i$. From (3), available bandwidth A is the minimum of all A_i ; thus we can induce that $R_{i-1} \geq R_i \geq A$. Bandwidth estimation tools such as IGI[8] and Spruce[10] are two examples that benefit from this observation.

$$A_i^2 - (C_i + R_{i-1})A_i + C_i R_{i-1} \geq 0 \quad (8)$$

C. OWD vs Packet Dispersion:

The relation between packet dispersion and OWD can be expressed briefly as in Fig. 1. When the probing rate is less than the available bandwidth, we will most-likely have $\Delta D^k = 0$ and $\Delta \delta = \delta_{out} - \delta_{in} = 0$. In other words, it means that there is no delay trend of OWD and the packet dispersions of the packet pair measured at the sender and the receiver would be the same. On the other hand, when the probing rate is more than the available bandwidth, the cross traffic can enlarge the dispersion, and it will cause the increasing of queuing delay. In summary, we can conclude that the packet dispersion and OWD are two criterions that can work together to estimate the available bandwidth.

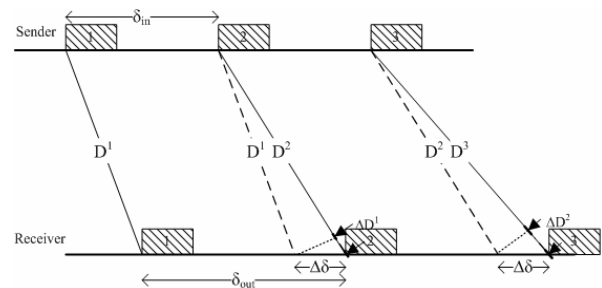


Fig. 1: The relationship between OWD and packet dispersion

Available bandwidth can fluctuate dramatically and thus it is very important for the bandwidth measurement to converge fast and accurately. In the previous bandwidth estimation algorithms, such as Pathload[3], which uses *binary search* to adjust the probing rate for the next iteration, and IGI[8], which updates the probing rate by some fixed step size to inspect whether the probing rate matches the available bandwidth, they might be too inefficient to infer the probing rate for the next iteration, especially in real-time distributed application, in addition to possible resolution issue of the estimated bandwidth.

III. PROPOSED METHOD

Before laying down the proposed algorithm, we use ns2 network simulator to conduct simulations with the topology

shown in Fig. 2 to observe our analysis shown in the previous section. The capacities along the path are 100, 75, 55, 40, 60, and 80 Mbps, respectively. Cross traffics are generated from 16 random sources at each link. The inter-arrival time of those cross traffics from each and every source follows Pareto distribution with exponential factor $\alpha=1.5$. A packet train transmitted by sender S consists of 10 packets of packet length $L=1500$ bytes with fixed packet dispersion δ_{in} so that the probing rate R_{in} ($=L/\delta_{in}=40\text{Mbps}$) for these 10 packets is greater than available bandwidth. The receiver R will record the dispersion δ_{out} of the arrival packets under different network utilizations of the bottleneck (i.e., the tight link).

In Fig. 3 we show the received packet dispersion distribution of the bottleneck link at various network utilizations. It is obvious that the received dispersion δ_{out} is influenced by the utilization of the tight link and has positive correlation with the network utilization when the fixed probing rate (R_{in}) is greater than the available bandwidth (A). The reason is that if the network load gets heavier, there is a higher probability that cross-traffic packets will be placed among the probing packets and they will contribute to the packet dispersion of the probing traffic. In addition, since the received probing rate R_{out} ($=L/\delta_{out}$) is inversely proportional to the received dispersion δ_{out} and also the available bandwidth A equals to $C-\lambda$, the received probing rate at R will have positive correlation to available bandwidth. Under this condition, the received probing rate is also an *upper bound* of the available bandwidth as shown in Section II, Part B. Thus, we can form the following equation:

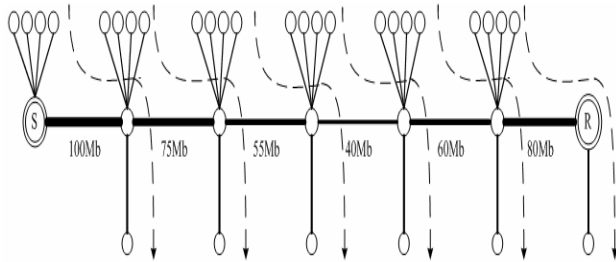


Fig. 2: Network topology used in the ns2 simulations.

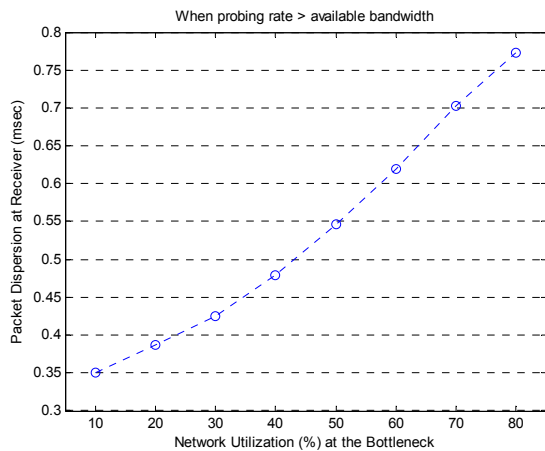


Fig. 3: The received packet dispersion of probing traffic at various traffic loads, given that the probing rate is greater than the available bandwidth.

$$R_{in} \geq R_{out} \geq A. \quad (9)$$

Based on the inequality in (9), we can have a better “guess” of the probing rate for the next iteration.

Initially, the probing sender transmits a packet train with probing rate (R_{in}) equal to the capacity of tight link or it can simply be a packet train with back-to-back packets. After passing through the links, its receiver observes the dispersions of the packet train and also the corresponding rate ($R_{out}=L/\delta_{out}$). Then it feedbacks the information R_{out} to the sender as the next probing rate. We continue above steps iteratively until the probing rate and available bandwidth start to match by means of performing the delay trend detection on the OWD. We modify the *full search algorithm* [5] to detect the delay trend. It is shown in [5] that full search algorithm is better than the Pairwise Comparison Test (PCT) and Pairwise Difference Test (PDT) of the delay trend detection in Pathload[3]. The proposed algorithm for available bandwidth estimation is showed in the following pseudo codes:

```

Proposed Available Bandwidth Estimation Algorithm:
{ /*initial*/
/* if Capacity of tight link is unknown, send back to back packet train */
If (Capacity of the tight link is known)
    Rin= capacity of the tight link;
Else
    Rin= rate of 10 back-to-back packets;
Rout= Get_Output_Rate_At_The_Receiver(Rin);
Dk= Get_Packet_Train_OWD (received packets at receiver);
While ( Is_Delay_Trend_Detected(Dk, Rin, Rout) )
{ /*if Rin>A, there will be delay trend*/
    Rin = Rout;
    Rout = Get_Output_Rate (Rin);
    Dk = Get_Packet_Train_OWD (received packets at receive);
}
Available Bandwidth = Rout; //goal
}
Bool Is_Delay_Trend_Detected ( Dk , Rin, Rout )
{
/*full search algorithm for delay trend detection*/

/* if Dk>Dl , then I(Dk>Dl)=1; if Dk<=Dl I(Dk>Dl)=0*/

$$S = \frac{\sum_{k=2}^N \sum_{l=1}^{k-1} I(D_k > D_l)}{M(M-1)}$$

/* delay trend detection and ε is to avoid vibration when probing rate
and the received rate are close enough.*/
If ( S > threshold or Rin-Rout < ε )
    return true;
else
    return false;
}

```

We use received probing rate as the next probing rate for the sender due to the analysis that the received probing rate has positive correlation with available bandwidth and it is also an upper bound of the available bandwidth. The estimation process can converge faster as shown in the next section. In addition, we use OWD as a criterion to decide whether probing rate is less than or equal to available bandwidth. Comparing with the other algorithms in literature, such as Pathload[3] or IGI[8], our algorithm is more robust. For

example, if the available bandwidth suddenly drops down during the estimation period, the Pathload or IGI may overestimate the available bandwidth and thus, it is not appropriate for multimedia streaming. However, our probing rate is based on the previous output rate; it will not be influenced by the sudden decrease of available bandwidth. On the other way, when available bandwidth increases during the probing, the estimation might be underestimated but it will not induce packets lost.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

A. Simulations

The simulation topology is the same as in Fig. 2 and the length of a packet train is 30 packets. We assume that the link with capacity 40 Mbps is the tight and narrow link and the queue length is 20 packets. Cross traffic is generated from 16 random sources at each link with Pareto distribution as stated earlier. To mimic the real Internet, cross traffic is simulated with different packet sizes in each stream as follows: 40% are 40 bytes, 50% are 550 bytes, and 10% are 1500 bytes. The threshold in the pseudo codes is 0.7 and ϵ is 2 Mbps.

We compare our algorithm with the binary-search based algorithm as in [5]. The available bandwidth is estimated every 20 sec., and the spots in Fig. 4 are the starting time of each estimation and corresponding available bandwidth estimated. From Fig.4, our algorithm keeps much closer to the curve of real available bandwidth.

Table I shows the mean absolute difference (MAD) between the estimated and true bandwidths. The required average numbers of packet trains to complete bandwidth estimation under different network utilization are also shown. Our algorithm outperforms binary-search based algorithm, especially in heavy traffic load.

B. Discussion

Using binary search to infer available bandwidth is unstable with only one threshold, because once the probing rate falls in the gray region [3], the results may be inconsistent. Even though Pathload uses two thresholds to detect delay trend, it is still easy to misjudge in the fast fluctuant network. Our proposed algorithm utilizes the features of OWD and dispersion to avoid above problems and to improve the accuracy and convergence speed of probing.

V. CONCLUSIONS AND FUTURE WORK

The proposed algorithm combines two features, i.e. OWD and packet dispersion, of short packet train to estimate available bandwidth from end-to-end hosts. Moreover, the proposed algorithm does not require to know the capacity of tight link in advance and it can start the probing by sending a back-to-back packet train initially. It is a simple but efficient algorithm which outperforms binary-search based algorithms and it is quite suitable for real-world multimedia streaming. The algorithm does not require synchronized clocks between the sender and its receiver for the delay trend detection to work. However, it may still have many challenges for us to

research in the future, such as clock skew and context switch during sending and receiving probing packets.

REFERENCES

- [1] C. Dovrolis, P. Ramanathan, and D. Moore, "Packet-dispersion techniques and a capacity- estimation methodology," IEEE/ACM Transaction on Networking, Vol. 12, NO. 6, Dec. 2004.
- [2] R. S. Prasad, M. Murray, C. Dovrolis and K.Claffy, "Bandwidth estimation: metrics, measurement techniques, and tools," Network, IEEE, pp. 27- 35, Vol. 17, Issue 6, Nov.-Dec. 2003.
- [3] M. Jain and C. Dovrolis, "Pathload: a measurement tool for end-to-end available bandwidth," in Proc. Passive Active measurements, Fort Collins, CO, Mar. 2002.
- [4] V. Ribeiro, R. Riedi, R. Baraniuk, J. Navratil, and L. Cottrell "pathChirp: efficient available bandwidth estimation for network paths," Passive and Active Measurement Workshop 2003.
- [5] Q. Liu and J. N Hwang, "End-to-end available bandwidth estimation and time measurement adjustment for multimedia QoS," International Conference on Multimedia and Expo (ICME), Jul. 2003.
- [6] B. Melander, M. Bjorkman, and P. Gunningberg, "A New End-to-End probing and Analysis method for estimating bandwidth bottlenecks," In Global internet Symposium, 2000.
- [7] M. Jain and C. Dovrolis, " End-to-end available bandwidth: measurement methodology, dynamics, and relation with tcp throughput," IEEE/ACM Transaction on Networking, Vol. 11, NO.4, Aug. 2003.
- [8] N. Hu, and P. Steenkiste, "Evaluation and characterization of available bandwidth probing techniques," IEEE Journal on Selected Areas in Communications, Vol. 21, No.6, Aug 2003.
- [9] X. Liu, K. Ravindran, and D. Longuinov, "What signals do packet-pair dispersion carry? "In Proceedings of IEEE INFOCOM, pp.281-292, Vol. 1, Mar. 2005.
- [10] J. Strauss, D. Katabi, and F. Kaashoek, "A measurement study of available bandwidth estimation tools," In Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement, pp.39-44, Oct. 2003.

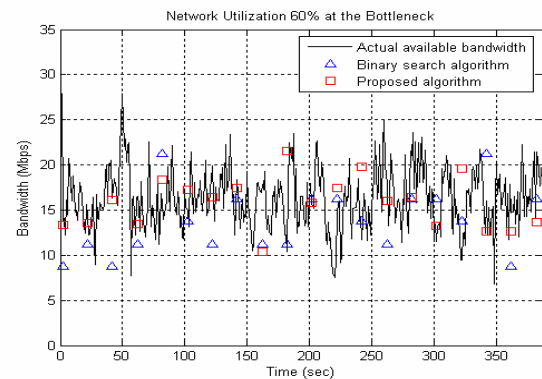


Fig. 4: The estimated available bandwidth in network utilization 60% (average available bandwidth is 16Mbps)

TABLE I. THE MAD AND PROBING NUMBER UNDER DIFFERENT NETWORK CONDITION

Network Utilization	Binary Search		Proposed algorithm	
	MAD (Mbps)	Probing Number	MAD (Mbps)	Probing Number
20%	1.98	4	1.19	2.25
40%	3.55	4	1.82	2.7
60%	3.62	4	2.27	3.05