DESIGN, IMPLEMENTATION AND EVALUATION OF ICMP-BASED AVAILABLE NETWORK BANDWIDTH MEASUREMENT BASED ON IMTCP

Hiroyuki Hisamatsu[1] and Hiroki Oda[2]

[1]Department of Computer Science, Osaka Electro-Communication University, Osaka, Japan
[2]Graduate School of Computer Science and Arts, Osaka Electro-Communication University, Osaka, Japan

ABSTRACT

We propose a method to measure available network bandwidth using the Internet Control Message Protocol (ICMP). The recently proposed ImTCP technique uses Transmission Control Protocol (TCP) data packets and the corresponding acknowledgement responses to measure the available bandwidth between sender and receiver. Since ImTCP needs to change the sender's TCP implementation, it needs modifications to sender's operating system kernel. Moreover, ImTCP cannot measure available bandwidth accurately if the receiver sends delayed acknowledgments. These problems stem from the use of TCP. In this paper, we discuss an ICMP-based method that overcomes these limitations. We evaluate the performance of the proposed method in an experimental network and show that it generates less measurement traffic and requires less time for bandwidth measurement than PathLoad. We also show that proposed method can measure the available bandwidth even if the bandwidth changes during measurement.

KEYWORDS

Available bandwidth, Bandwidth Measurement, Inline measurement TCP (ImTCP)

1. Introduction

In recent years, Internet-based services have proliferated with the increase in network speed and the number of Internet users. We now have various network services based on peer-to-peer (P2P) networks [1], contents delivery networks (CDN) [2], grid networks [3], and IP-VPN [4]. These network services build original logical networks, called overlay network on top of the Internet Protocol (IP) network. In order to enhance the service qualities in an overlay network, it is essential to understand the resource status of the IP network and to utilize the network effectively. Network bandwidth is the most essential resource status parameter. When multiple peers hold the same resource in a P2P network, we can use network bandwidth information to choose a peer to retrieve the resource from. In a CDN, we can transfer data at low priority on the basis of bandwidth information so as not to affect high priority data transfer [5]. Network bandwidth information can also be used for the determination of the failure point in a network.

DOI : 10.5121/ijcnc.2014.6301

Moreover, the video streaming services such as YouTube [6], Dailymotion [7], and Veoh [8] have become more and more popular. Much research has been conducted on new transport layer protocols for video streaming [9,10,11,12,13,14,15,16,17]. For instance, in [17], a new transport-layer protocol, called TCP Stream, for video streaming have been proposed. As the results of our simulations, it have been shown that when a network is in a congestion state, TCP stream transmits data packets at an adjusted rate required for the video sequence, unlike TCP NewReno, and does not steal bandwidth from other network traffic. The network bandwidth is very importance for video streaming service.

Much research has already conducted on network bandwidth measurement, and indices of network bandwidth such as physical bandwidth, available bandwidth, and bulk transfer capacity (BTC) have been defined [18,19]. Physical bandwidth refers to the bandwidth of the bottleneck link when no other traffic is competing on the network path between sender and receiver. An available bandwidth refers to the bandwidth that competing traffic does not use. BTC is the data transfer throughput after a sufficiently long time following the initiation of data transfer. Many methods to measure these indices have been developed [20,21,22].

However, since above-mentioned methods are time-consuming, it is difficult to use them for real-time measurement. In addition, since they require the transmission of many packets over a network, bandwidth measurement can severely affect the network. For instance, PathLoad [18] transmits many packets in a short period over a network to measure available bandwidth, causing congestion. There are also some methods of measuring available bandwidth that require cooperation between the routers and end-hosts in a network [23,24]. These methods can measure the available bandwidth of the network using only a small number of packets, but they need to modification of all the routers in the network.

Inline measurement TCP (ImTCP) is a recently developed method for measuring available bandwidth [25]. ImTCP utilizes only TCP data packets and the corresponding acknowledgement (ACK) packets to measure the available bandwidth between the sender and receiver. Since ImTCP needs to modify sender-side TCP, it needs to modify the operating system kernel at the sender host. Therefore, ImTCP cannot be installed on operating systems whose kernel cannot be modified by a user such as MS Windows. Moreover, when a receiving host's delayed ACK option is in effect, ImTCP cannot obtain exact bandwidth measurements.

In this paper, we propose a new method to measure available bandwidth. Based on the ImTCP algorithm, the proposed method measures available bandwidth using ICMP ECHO packets and ICMP ECHO REPLY packets and does not require modification to a receiver. Further, the proposed method does not need modifications to the operating system kernel at the sender host and can thus be installed on MS windows. When a receiver host receives an ICMP ECHO packet, an ICMP ECHO REPLY packet is sent immediately. Therefore, the proposed method works even when the delayed ACK option is in effect at the receiver host. In this paper, we describe the implementation of our method, and evaluate its performance on an experimental network. Note that this paper is an extended version of work published in [22]. We extend our previous work by the method design and the performance evaluation.

The rest of this paper is organized as follows. In Section 2, we explain the ImTCP algorithm. In Section 3, we explain the design of our method, and describe its implementation. In Section 4, we show the effectiveness of the proposed method by conducting a performance evaluation on an experimental network. Finally, in Section 5, we state our conclusions and discuss future work.

2. INLINE MEASUREMENT TCP

In this section, we describe the ImTCP measurement algorithm and its limitations.

2.1. ImTCP Measurement algorithm

ImTCP measures the available bandwidth between a sender and receiver hosts. Figure 1 shows an outline of ImTCP bandwidth measurement. In TCP data transfer, a sender transmits data packets to a receiver host, and the receiver host transmits ACK packets to the sender if these data packets have been appropriately received. ImTCP utilizes this TCP mechanism, adjusting transmission intervals at the sender host and observing the changes in the arrival intervals of the ACK packets. The available bandwidth is calculated on the basis of this information.

PathLoad, a tool for available bandwidth measurement, requires a sender to transmit multiple measurement packets to a receiver host at a constant interval. A receiver host measures available bandwidth by observing the change in the arrival interval of the received packets. If the transmission and arrival intervals are equal, it is assumed that the transmission rate of the packets for measurement is lower than the available bandwidth, and PathLoad increases the transmission rate. If the arrival interval is greater than the transmission interval, it is assumed that the transmission rate is higher than the available bandwidth, and PathLoad reduces the transmission rate. A bisection search gives the available bandwidth. Many packets are required to be sent for measurement by PathLoad.

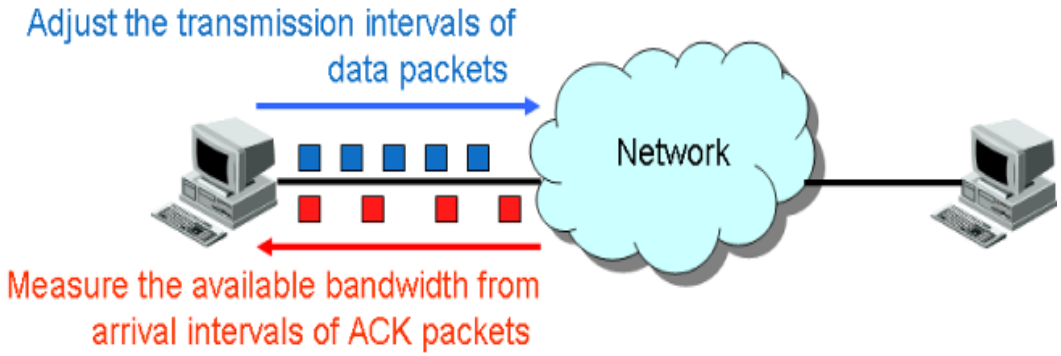

Figure 1. Outline of ImTCP Mechanism

ImTCP measures available bandwidth by changing transmission intervals of TCP data packets and observing the arrival intervals of the corresponding ACK packets. ImTCP requires fewer packets for measurement than constant-interval transmission methods such as PathLoad. ImTCP uses past measurement results to estimate a range for the current available bandwidth (called "search range" hereafter) and searches within this range. If the available bandwidth changes rapidly, it may no longer lie with in the search range; in this case, the measurement algorithm of ImTCP configures a new search range based on the newest measurement result. Thus, ImTCP can find the correct value even if it lies outside the search range. ImTCP operates as follows:

1. Perform the first measurement based on the Cprobe algorithm [20].
2. Determine an initial search range based on the measurement result.

3. Divide the search range into k sub-ranges.
4. Transmit n packets for measurement of each sub-range.
5. Observe the arrival intervals of ACK packets for each sub-range.

   On the basis of the arrival intervals, choose a sub-range that contains the available bandwidth.
6. Calculate the available bandwidth from the chosen sub-range.

   Derive a 95% confidence interval from past measurement results, and set this intervals as the next search range.
7. Return to (3).

Please refer to [13] for a detailed description of ImTCP.

2.2. ImTCP limitations

As described, ImTCP utilizes the TCP acknowledgment mechanism. If delayed ACK is in effect, when a data packet arrives at a receiver host, the receiver host does not transmit an ACK response immediately but delays it until one of the following conditions is satisfied.

1. A data packet is transmitted from receiver to sender.
2. Another data packet is received from the sender.
3. 200 [ms] have passed since the arrival of the previous packet.

Thus, if the TCP delayed ACK option is in effect at a receiver host, ImTCP cannot measure the available bandwidth accurately. We note that the TCP delayed ACK is enabled by default in Windows 7, Mac OS X Snow Leopard, and Linux kernel version 2.6.

Moreover, in a sender host, ImTCP buffers the data packets in order to adjust the transmission interval. Therefore, in order to install ImTCP, it is necessary to modify the operating system kernel at the sender host. ImTCP cannot be installed on an operating system whose kernel cannot be modified by the user.

3. BANDWIDTH MEASUREMENT METHOD USING ICMP

In this section, we illustrate the design of the ICMP-based available bandwidth measurement method and explain its implementation.

3.1. Design

In the proposed method, we measure available bandwidth by using ICMP ECHO packets and ICMP ECHO REPLY packets to overcome ImTCP's delayed-ACK limitation. A receiver host returns an ICMP ECHO REPLY packet immediately upon receiving an ICMP ECHO packet. Therefore, a sender host can always observe the arrival interval of ICMP ECHO REPLY packets. Moreover, we can transmit ICMP ECHO packets from an application at arbitrary times. Therefore, we can adjust the transmission interval without modifying the operating system kernel at the

sender host. Therefore, the proposed method can be utilized in an operating system whose kernel is not user-modifiable.

The measurement algorithm of ImTCP searches for available bandwidth in a restricted search range. Therefore, the number of packets required for measurement of the proposed method is smaller than that for existing measurement methods. Based on the measurement algorithm of ImTCP, the proposed method also requires fewer packets for measurement than the existing measurement methods.

The proposed method terminates measurement when the search range, which contains the required value of available bandwidth, becomes sufficiently small. Let *upper* denote the upper limit of a search range, *lower* denote the lower limit, and *abw* denote the last measurement result. When the following inequality is satisfied, the proposed method terminates measurement and shows the current value of *abw* as the available bandwidth.

$$upper - lower < \alpha \times abw$$

ICMP is abused in many cases as means of attacking server, as in the ping flood attack [27] where an aggressor brings a server down by transmitting ICMP ECHO packets in large quantities to the server. When a lot of ICMP ECHO packets are received in a short time, many servers are configured to filter out these packets. Therefore, when measuring available bandwidth using ICMP, we must limit the number of ICMP ECHO packets transmitted. In the proposed method, we set an idle period (during which measurement is not conducted) after each measurement that is equal to two round-trip durations. This prevents the transmission of a lot of ICMP ECHO packets in a short time. Moreover, setting the idle period eliminates the effect of the packets transmitted for the last measurement on the network.

Table 1. HZ values, clock resolution, and measurable bandwidth

HZ	Clock Resolution [μs]	Measurable Bandwidth [Mbit/s]
100	10,000	0.8
250	4,000	2
1,000	1,000	8
10,000	100	80
20,000	50	160
50,000	20	400
100,000	10	800

3.2. Implementation

The proposed method transmits ICMP ECHO packets with timings based on the ImTCP. We implement packet sending through a select() system call. The granularity of a select() system call depends on the time granularity of the kernel. Therefore, we cannot configure the transmitting interval of ICMP ECHO packets to a smaller interval than the time granularity of the kernel.

The time granularity of a kernel is determined by the kernel constant HZ in Linux. Table 1 shows HZ values with corresponding time granularities and the measurable bandwidth when a packet interval is adjusted based on the time granularity and the packet size is 1000 [byte]. In the Linux kernel, current default HZ is 250, corresponding to a time granularity of 4000 [μs] from Table 1. Thus, the maximum bandwidth that we can measure is 2 [Mbit/s]. In order to measure the available bandwidth in a high-speed network, it is necessary to increase the value of HZ.

However, if the value of HZ is increased, task switching may occur frequently and the associated overheads may affect the execution speed of the kernel [28]. After careful consideration, we set HZ to 50000 in our experiment.

4. PERFORMANCE EVALUATION

In this section, we evaluate the proposed method in an experimental network and show its effectiveness.

4.1. Experimental Network

Figure 2 shows the experimental network, which is constructed from a PC router running DummyNet, a traffic generator, a sender, and a receiver. Table 2 shows the specifications of computers used in the experimental network. ImTCP estimates the tendency of the receiving interval of measurement packets using two thresholds, PCT and PDT. In this paper, PCT is configured to 40 and PDT is configured to 30. Further, we set the number of subdivisions of a search range K to four, the number of packets for measuring the sub-range n to 10, and the parameter for end of measurement α to 0.05. The parameters used are summarized in Table 3.

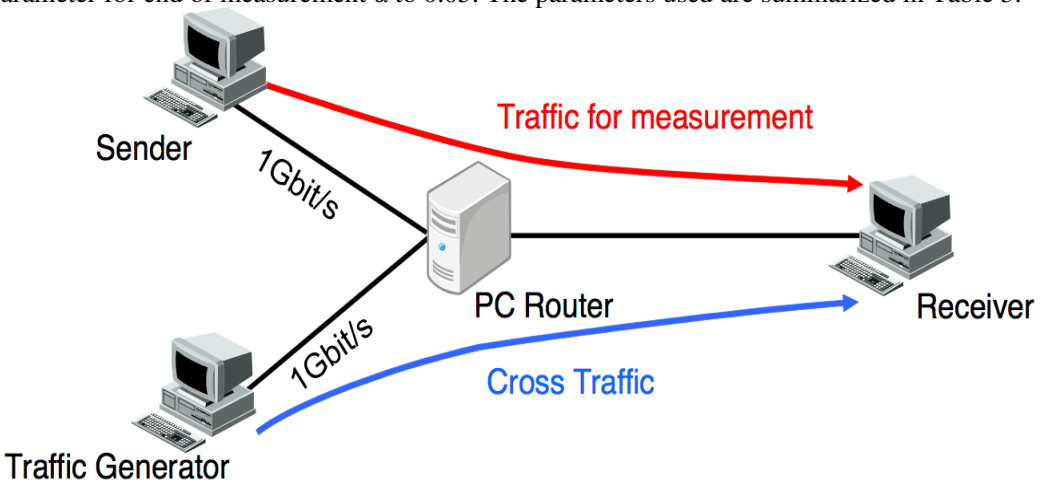

Figure 2. Experimental network

Table 2. Computer specifications in the experimental network

Host	Sender & Receiver	PC Router	Traffic generator
CPU	Core2Duo 3 [GHz]	Core2Quad 3 [GHz]	CoreDuo 1 [GHz]
Memory	3 [GByte]	16 [GByte]	1.5 [GByte]
OS	Fedora Core 5	FreeBSD 7.0	Ubuntu 9.10

Table 3. Parameter settings for the proposed method

Parameter	Value
HZ	50,000
PDT	40
PCT	30
K	4
N	10
α	0.05

4.2. The amount of data and time required for measurement

The measurement methods transmit packets over a network for measurement. In other words, they give load to a network. The amount of data used for measurement is an essential index of the measurement methods. In addition, compared with the physical bandwidth, the time granularity of the available bandwidth variation is very small. Therefore, the time for measurement is important. We first evaluate the amount of data and time required for measurement. In this evaluation, the bandwidth between the PC router and the receiver is varied from 1 [Mbit/s] to 5 [Mbit/s] using DummyNet. We took 10 readings at each setting. We also took readings using PathLoad for comparison.

Figure 3 shows the distribution of the relative error between measured available bandwidth and actual available bandwidth and the total amount of data for measurement sent to the receiver. From Figure 3, it is observed that the measurement error of the proposed method is 0.1 or less, much smaller than that of PathLoad. The minimum and maximum amounts of data transmitted by the proposed system are 162 [Kbyte], and 1134 [Kbyte], respectively. When using PathLoad, the corresponding values are 2163 [Kbyte] and 3324 [Kbyte]. In most cases, it is observed that the proposed method transmits only a small amount of data for measurement. Even in the worst case, the total amount of data that the proposed method requires for measurement is 1134 [Kbyte]. We note that the minimum amount of data required by PathLoad is 2163 [Kbyte].

Figure 4 shows the distribution of the relative errors between measured available bandwidth and actual available bandwidth and the measurement end time. It is observed that the proposed method can measure available bandwidth in 3–4 [s], much faster than PathLoad. Even when time is taken to finish the measurement, the proposed method takes only 30 [s].

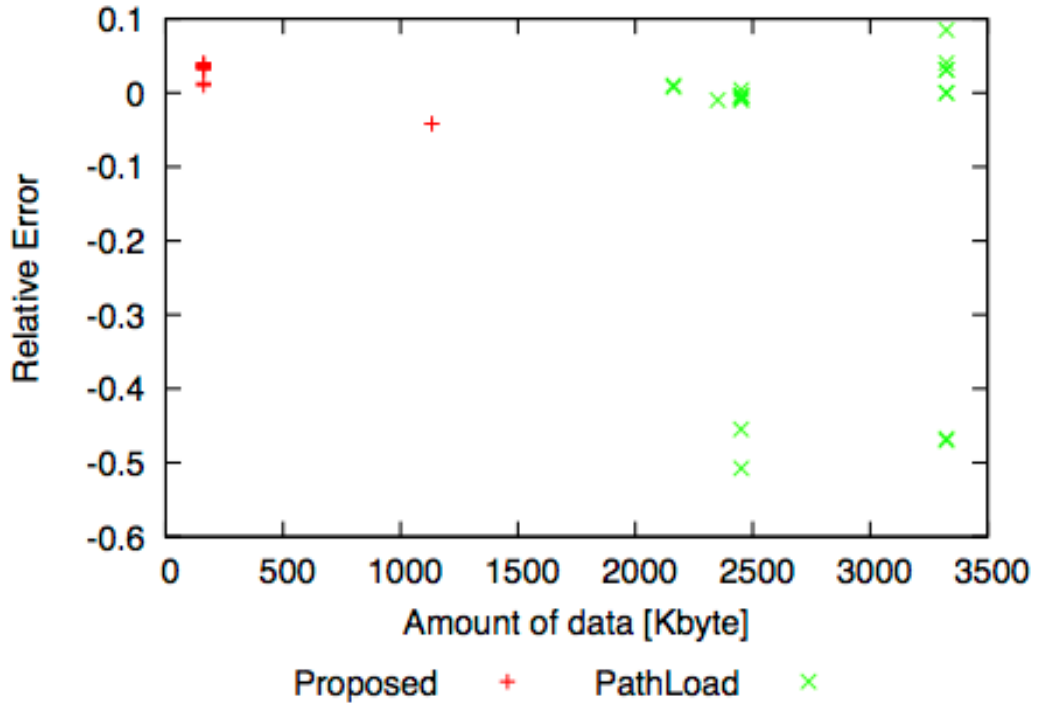

Figure 3. Distribution of the relative errors between measured and actual bandwidth and the total amount of data

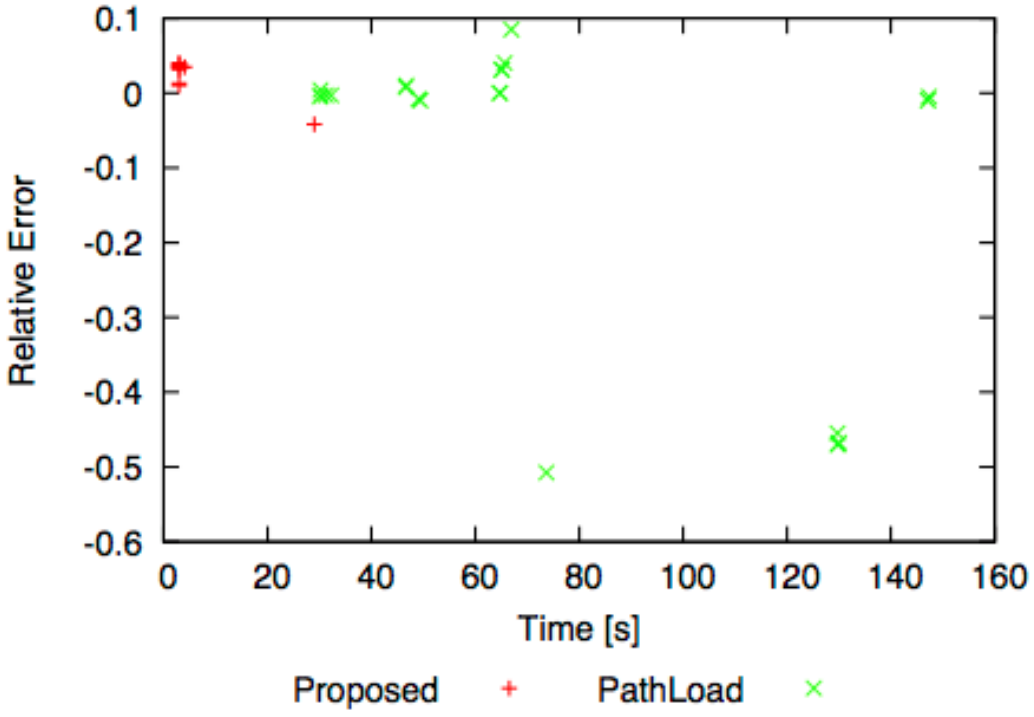

Figure 4. Distribution of the relative errors between measured and actual bandwidth and the measurement time

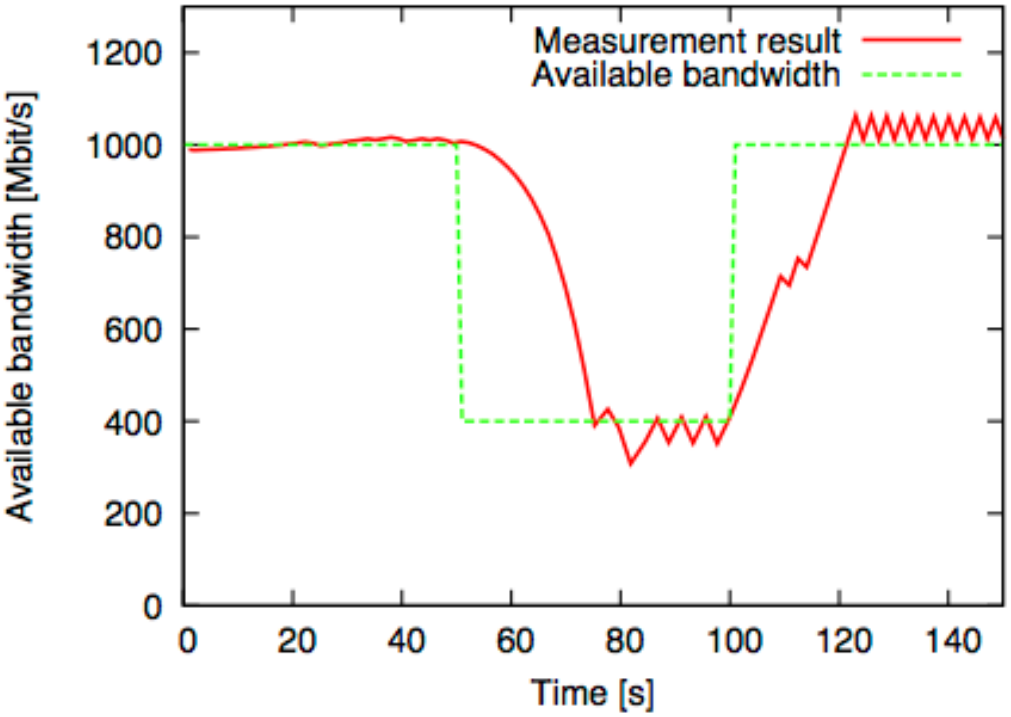

Figure 5. Tracking bandwidth changes in narrow-bandwidth network

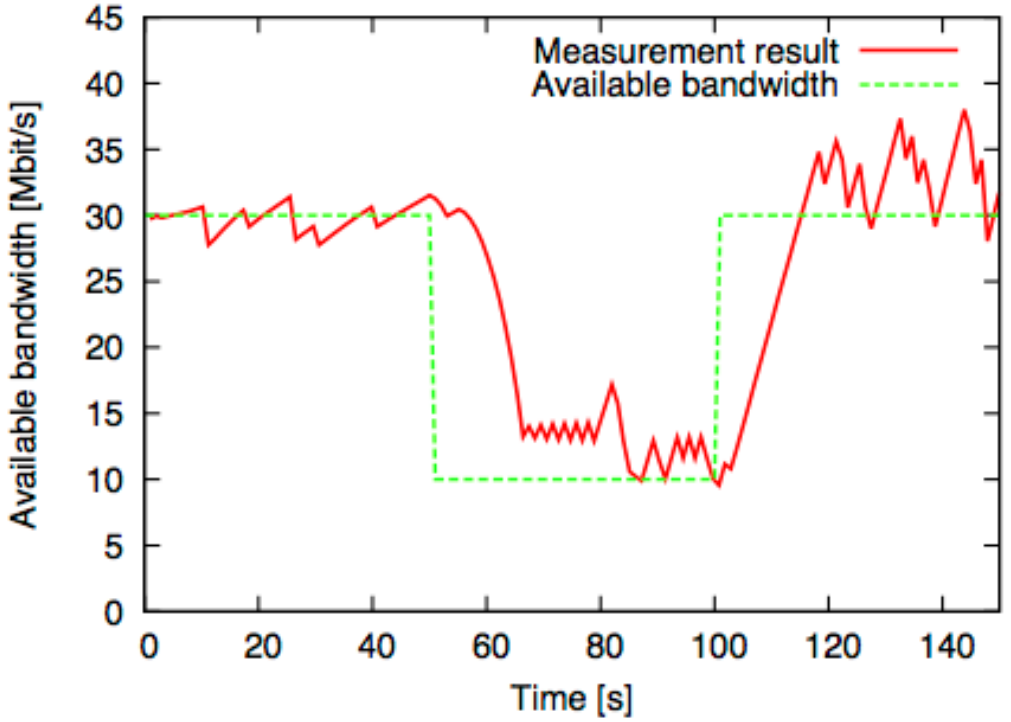

Figure 6. Tracking bandwidth changes in high-speed network

4.3. The tracking for the change of the available bandwidth

Next, we evaluate how the proposed method tracks changes in available bandwidth in a narrow-band network and a high-speed network. To simulate a narrow-band network, the bandwidth between the PC router and the receiver is configured to 1 [Mbit/s] using DummyNet. 50 [s] after starting the experiment, we use Iperf [29] to generate UDP cross traffic of 600 [Kbit/s] for 50 [s]. To simulate a high-speed network, the bandwidth between the PC router and the receiver is configured to 30 [Mbit/s] using DummyNet. 50 [s] after starting the experiment, Iperf is used to generate UDP cross traffic of 20 [Mbit/s] for 50 [s].

The change in the actual available bandwidth and the measurement results are shown in Figure 5 and 6. Figure 5 shows that the proposed method follows the change of the available bandwidth after a 20 [s] delay. This is because the proposed method searches within the search range so as to avoid transmitting many packets in a short period. The proposed method obtains the next search range using past measurement results. If the measurement result is stable, the proposed method narrows the next search range. Since the available bandwidth changes after the search range becomes small, the tracking is delayed. Figure 6 also shows the result following the change of available bandwidth after a 20 [s] delay. The reason for the delay is the same as that for the narrow-band network.

From the evaluation results, we observe that the proposed method can measure the available bandwidth using less data than PathLoad, which is the conventional available bandwidth measurement method. In addition, it is shown that proposed method can measure the available bandwidth faster than PathLoad. Further, we show that the proposed method experiences a delay in tracking the available bandwidth when the available bandwidth changes.

5. CONCLUSION AND FUTURE WORK

In this paper, we proposed a new available bandwidth measurement method based on the measurement algorithm of ImTCP, an inline network measurement method. The proposed method solves the limitations inherent to ImTCP. We evaluated the proposed method in an experimental network, and showed that the proposed method requires smaller amounts of data and less measurement time compared to PathLoad. Further, although the proposed method suffers from tracking delays when there is a change in available bandwidth, it does successfully measure the available bandwidth even under such circumstances.

As future work, we intend to conduct an investigation into the parameter configuration of the proposed method compared to that of ImTCP. This may provide further insight into the tracking delay.

ACKNOWLEDGEMENTS

This work was partly supported by Dayz Inc.

REFERENCES

[1] A. Rao, K. Lakshminarayanan, S. Surana, R. Karp, and I. Stoica, "Load balancing in structured P2P systems," in Proceedings of International Workshop on Peer-to-Peer Systems(IPTPS 2003), pp. 68–79, Feb. 2003.

[2] J. Dilley, B. Maggs, J. Parikh, H. Prokop, R. Sitaraman, and B. Weihl, "Globally distributed content delivery," IEEE Internet Computing, vol. 42, pp. 50–58, Oct. 2002.

[3] Y. Zhao and Y. Hu, "GRESS - a grid replica selection service," in Proceedings of International Conference on Parallel and Distributed Computing and Systems(PDCS 2003), pp. 423–429, Aug. 2003.

[4] S. Jha and A. Sood, "An architectual framework for management of IP-VPNs," in Proceedings of the 3rd Asia-Pacific Network Operations and Management Symposium, Sept. 1999.

[5] Tsugawa, G. Hasegawa, and M. Murata, "Background TCP data transfer with inline network measurement," in Proceedings of Asia-Pacific Conference on Communications(APCC 2005), pp. 459–463, Oct. 2005.

[6] "Youtube." available at http://www.youtube.com/.

[7] "Dailymotion." available at http://www.dailymotion.com/.

[8] "Veoh." available at www.veoh.com/.

[9] L. Cai, X. Shen, J. Pan, and J. Mark, "Performance analysis of TCP-friendly AIMD algorithms for multimedia applications," IEEE/ACM Transactions on Networking, pp. 339–355, Apr. 2005.

[10] F. Yang, Q. Zhang, W. Zhu, and Y.-Q. Zhang, "End-to-end TCP-friendly streaming protocol and bit allocation for scalable video over wireless Internet," IEEE Journal on Selected Areas in Communication, pp. 777–790, May 2004.

[11] S. L. Bangolae, A. P. Jayasumana, and V. Chandrasekar, "TCP-friendly congestion control mechanism for an UDP-based high speed radar application and characterization of fairness," in Proceedings of the The 8th International Conference on Communication Systems, pp. 164–168, Nov. 2002.

[12] Y.Zhu, A.Velayutham, O.Oladeji, and R.Sivakumar, "Enhancing TCP for networks with guaranteed bandwidth services," ACM Comuputer Networks: The International Journal of Computer and Telecommunications Networking, vol. 10, pp. 2788–2804, July 2007.

[13] T.Tsugawa, N.Fujita, T.Hama, H.Shimonishi, and T.Murase, "TCP-AFEC: An adaptive FEC code control for end-to-end bandwidth guarantee," in Proceedings of 16th International Packet Video Workshop (PV2007), Nov. 2007.

[14] H. Hisamatsu, G. Hasegawa, and M. Murata, "Non bandwidth-intrusive video streaming over TCP," to be presented at 8th International Conference on Information Technology : New Generations (ITNG 2011), Apr. 2011.

[15] R. Kuschnig, I. Kofler, and H. Hellwagner, "Improving Internet video streamilng performance by parallel TCP-based request-response streams," in Proceedings of the 7th IEEE Conference on Consumer Communications and Networking Conference, pp. 200–204, Jan. 2010.

[16] G. S. Rao, P. S. Kumar, K. D. Prasad, M. Supraja, G. V. Kumar, V. S. V. S. Murthy, M. P. Kumar, D. Rajesh, S. Lavanya, P. A. Deepthi, and C. P. Satish, "Architecture for efficiently streaming stored video using TCP," International Journal of Computer Science and Network Security, vol. 10, pp. 154–163, Jan. 2010.

[17] H. Oda, H. Hisamatsu, and H. Noborio, "Design, implementation and evaluation of congestion control mechanism for video streaming," International Journal of Computer Networks & Communications (IJCNC), vol. 3, pp. 193–204, May 2011.

[18] M. Jain and C. Dovrolis, "End-to-end available bandwidth: Measurement methodology, dynamics, and relation with TCP throughput," in Proceedings of ACM SIGCOMM 2002, pp. 295–308, Aug. 2002.

[19] M. Mathis and M. Allman, "A framework for defining empirical bulk transfer capacity metrics," RFC 3148, July 2001.

[20] R. L. Carter and M. E. Crovella, "Measureing bottleneck link speed in packet switched networks," International Journal on Performance Evaluation, vol. 27-28, pp. 297–318, Mar. 1996.

[21] A. Shriram, M. Murray, Y. Hyun, N. Brownlee, A. Broido, M. Fomenkov, and K. Claffy, "Comparison of public end-toend bandwidth estimation tools on high-speed links," in Proceedings of Passive and Active Measurement Workshop(PAM 2005), pp. 306–320, Mar. 2005.

[22] Q. He, C. Dovrolis, and M. Ammar, "On the predictability of lagrge transfer TCP throughput," in Proceedings of ACM SIGCOMM 2005, pp. 145–156, Aug. 2005.

[23] T. Anjali, C. Scoglio, J. C. D. Oliveira, L. C. Chen, I. F.Akyildiz, J. A.Smith, G. Uhl, and A. Sciuto, "A new path selection algorithm for MPLS networks based on available bandwidth estimation," in Proceedings of the 3rd international conference on quality of future internet services and internet charging and Qos technologies 2nd international conference on From QoS provisioning to QoS charging, pp. 205–214, Oct. 2002.

[24] S. Ubik, D. Antoniades, and A. Oslebo, "ABW - short-timescale passive bandwidth monitoring," in Proceedings of the Sixzth International Conference on Networking, p. 49, Apr. 2007.

[25] C. L. T. Man, G. Hasegawa, and M. Murata, "A simultaneous inline measurement mechanism for capacity and available bandwidth of end-to-end network path," IEICE Transactions on Communications, vol. E89-B, pp. 1614–1626, Sept. 2006.

[26] H. Oda, H. Hisamatsu, and H. Noborio, "Developing an ICMP-based network bandwidth measurement," in Proceedings of The Sixth International Conference on Mobile Computing and Ubiquitous Networking (ICMU2012), pp. 128–129, May 2012.

[27] J. Elliott, "Distributed denial of service attacks and the zombie ant effect," IP Professional, vol. 2, pp. 55–57, Apr. 2000.

[28] T. Tsugawa, G. Hasegawa, and M. Murata, "Implementation and evaluation of an inline network measurement algorithm and its application to TCP-based service," in Proceedings of IFIP/IEEE Workshop on End-to-End Monitoring Techniques and Services(E2EMON 2006), pp. 64–74, Apr. 2006.

[29] NLANR/DAST, "Iperf - the TCP/UDP bandwidth measurement tool." available at http://iperf.sourceforge.net/.

Authors

Hroyuki Hisamatsu received M.E. and Ph.D. degrees from Osaka University, Japan, in 2003 and 2006, respectively. He is currently an associate professor of Department of Computer Science, Osaka Electro communication university. His research work is in the the area of performance evaluation of TCP/IP networks. He is a member of IEEE and IEICE.

Hiroki Oda received B.E and M.E. degrees from Osaka Electro-Communication University, Japan, in 2009 and 2011, respectively. He is currently a doctoral student at the Graduate School of Computer Science and Arts, Osaka Electro-Communication University. His research interests include network performance evaluation, TCP protocol design and evaluation. He is a student member of IEICE.