Detection of Multiple Bottleneck Bandwidth

Niwat Thepvilojanapong
Graduate School of Information Science and Technology
The University of Tokyo
wat@mcl.iis.u-tokyo.ac.jp

Yoshito Tobe
Graduate School of Engineering
Tokyo Denki University
tobe@im.dendai.ac.jp

Kaoru Sezaki
Institute of Industrial Science
The University of Tokyo
sezaki@iis.u-tokyo.ac.jp

Abstract

*This paper endeavors to present a scheme to detect and estimate bottleneck bandwidth along the path in the Internet. We have participated in the RIPE NCC's TTM project to perform one-way delay (*OWD*) and loss measurement from a host in our laboratory to other hosts in Europe and USA. TTM is an active measurement system, which has implemented the IPPM one-way delay (*RFC2679*) and one-way loss metrics (*RFC2680*). From measured delay, loss, and* traceroute*'s data, we can know the path properties such as bandwidth, path rerouting, congestion between each host, and so on. Based on measured delay, we propose an algorithm, called* Estimating Bottleneck Bandwidth using Packet-pair (EBBP)*, to estimate bottleneck bandwidth. Our algorithm is based on Bolot's equation, but we use* OWD *instead of round trip delay. Every participated host uses GPS receiver to avoid the problem of clock difference. We make phase plot graph from measured delay, extract useful samples, quantize extracted samples, and find intercept of phase plot graph by* EBBP*. Finally, we can estimate bottleneck bandwidth along the path.*

1. Introduction

Network traffic measurement has been considered as a necessary activity since the early days of networking. Accurate measurement of network characteristics is important to a variety of network applications. Unfortunately, accurate measurement and estimation are difficult because of heterogeneity of today's Internet.

There are many useful network characteristics such as delay, loss, bandwidth and so on. Understanding these characteristics is important for the proper design of network algorithms such as routing and flow control algorithms, congestion detection algorithms [18], for the dimensioning of buffers and link capacity, for choosing parameters in simulation and analytic studies, and for proving and improving network topology generators [17]. It is also essential for designing the emerging audio and video applications. For example, the shape of the delay distribution or IP-delay variation (IPDV or jitter) is crucial for the proper sizing of playback buffers [6]. End-to-end delay or one-way delay (*OWD*) as well as loss of packets affects the performance of applications over the Internet.

Bandwidth can be divided into two different types: bottleneck bandwidth and available bandwidth. The bottleneck bandwidth of a route is the ideal bandwidth of the lowest bandwidth link (the bottleneck link or the narrowest link) on the route between two hosts. In most networks, as long as the route between two hosts remains the same, the bottleneck bandwidth remains the same. It is not affected by other traffics. In contrast, the available bandwidth of a route is the maximum bandwidth at which a host can transmit at a given point in time along that route – that is, the portion of bottleneck bandwidth that is not used by competing traffic. Available bandwidth is limited by other traffics along that route [13].

We can divide measurement techniques into two main methods: the passive and the active. Passive method uses existing traffic in the network, on the other hand, active method sends probe packets to the network for measurement. Methods for discovering network characteristics using measurements taken at endpoints are increasingly valuable as applications and services seek to adapt to network properties. There are many previous works on estimation of network properties, which have focused on estimating bottleneck bandwidth [6], [5], [8], [13], [16], [7], [10], estimating available bandwidth [7], [12], and estimating per-link bandwidth, latency, and loss [14], [11], [15], [9]. We perform active measurement to measure one-way delay and one-way loss according to the IPPM one-way delay (*RFC2679*) and one-way loss (*RFC2680*) metrics. From measured delay, loss, and *traceroute*'s data, we can know path properties such as bandwidth, path rerouting, congestion between each host, and so on. Based on measured delay, we propose an algorithm, called *Estimating Bottleneck Bandwidth using Packet-pair (EBBP)*, to estimate bottleneck bandwidth along the path. Our algorithm is based on Bolot's equation [6], [5], but we use *OWD*s instead of round trip delays to avoid the problem of asymmetric path. Previous works [16], [10] which measured *OWD*s had problems about clock difference and clock skew between each machine. Every host in our measurement uses GPS receiver to avoid such problems. Then we make phase plot graph from measured *OWD*, extract only useful samples, quantize extracted samples, and calculate intercept of phase plot graph by finding the closely clustered line. Finally, we can estimate bottleneck bandwidth of the path from known intercept and Bolot's equation. We also discuss changes in transmission rate to receive desired detectable range of bandwidth.

The remainder of the paper is organized as follows. Section 2 summarizes the basis in packet-pair model, also including the previous work on bottleneck bandwidth estimation. Section 3 describes RIPE's Test Traffic Measurement system. Section 4 presents a scheme for estimating bottleneck bandwidth. Section 5 shows results from our measurement. There are delay measurement and bottleneck bandwidth estimation from some pairs of participated hosts. Finally, we conclude this paper in section 6 and describe future works in section 7.

2. Packet-pair model

According to previous works [6], [5], [13], [16], [7], [10], packet-pair model has been used to estimate bottleneck bandwidth. This model may be the most popular measurement algorithm today because there are many tools that were developed by this concept and we can measure traffic by active and/or passive method. This model finds the difference in arrival times of two packets of the same size traveling from the same source to the same destination. If two packets are sent close enough together in time to cause two packets to queue together at the bottleneck link ($t_i^1 - t_i^0 < s/B_b$; variable definition is shown in Table 1), then two packets will arrive at the destination with the same spacing as when they exited the bottleneck link ($t_n^1 - t_n^0 = t_{i+1}^1 - t_{i+1}^0$; if there are n links along the path). The concept of packet-pair model is shown in Fig. 1 (reproduced from [14]). Therefore, we can estimate bottleneck bandwidth by using equation 1. Recent work [8] uses a number of packet-pairs, calculates bottleneck bandwidth according to equation 1, and estimates possible bottelneck bandwidth from distribution of calculated bottleneck bandwidth.

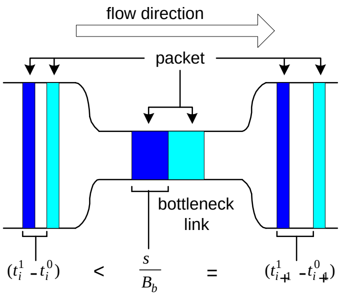

Figure 1. Packet-pair model

$$B_b = \frac{s}{t_n^1 - t_n^0} \tag{1}$$

Table 1. Variable definition

variable	definition
s	size of packet (bits)
B_b	bottleneck bandwidth (bits/second)
t_l^k	time when packet k fully arrives at link l (sec)
δ	transmission interval (second)

There is an assumption in the packet-pair model that transmission delay is linear with respect to packet size and that routers are store-and-forward. Another assumption is that the bottleneck router uses FIFO-queuing. If the router uses fair queuing, then packet-pair measures the available bandwidth of the bottleneck link. The advantages of packet-pair over the other techniques are that it measures the true bandwidth of the network (unlike throughput[1]), it does not cause packet loss, and it does not require many packet round trips to work or send massive amounts of data (unlike *pathchar* [11]). On the other hand, there are some of the key problems with current packet-pair algorithm: queuing failure, competing traffic, dropped probe packet, and downstream congestion. We can use filtering method or other techniques to deal with such problems.

2.1. Bolot's technique

Bolot [6], [5] uses the measured round trip delays of small UDP probe packets ($s = 32 \times 8$ bits) sent at regular time intervals ($\delta = 8, 20, 50, 100, 200,$ and 500 ms) to

---

[1] Throughput is the amount of data that a transport protocol like TCP can transfer per unit of time.

analyze the end-to-end packet delay and loss behavior in the Internet. He lets the source host be the same as the destination host to avoid clock difference between two machines. Then he compares the *RTT*s of adjacent packets from a sequence sent at regular intervals. The *RTT*s of successive packets are plotted against each other ($t_n^k \ vs \ t_n^{k-1}$), called *phase plot graph*. After that he analyzes measured delays and estimates bottleneck bandwidth.

Large value of δ is used for light load situation (equation 2). ϵ_n is a random process with mean 0 and low variance. The point in the phase plane should be scattered around the diagonal (the line $rtt_{n+1} = rtt_n$). On the other hand, small value of δ is used for heavy load situation (equation 3). he estimated bottleneck bandwidth from calculated intercept ($\delta - s/B_b$) of phase plot graph.

$$\begin{array}{rrl} \text{Light load:} & rtt_{n+1} = & rtt_n + \epsilon_n \qquad (2) \\ \text{Heavy load:} & rtt_{n+1} = & rtt_n - (\delta - s/B_b) \qquad (3) \end{array}$$

2.2. Huang's technique

Huang's algorithm [10] is mainly based on Bolot. He proposed technique to find the intercept from phase plot graph. The step of algorithm is: (1) extracting packets at the upper-right part of the graph by using twice the minimum delay as threshold in extraction; (2) quantizing extracted samples; and (3) estimating the intercept by finding a line where many markers concentrate in. In his work, using *OWD*s instead of *RTT*s leads to the possibility of decrease in cross traffics.

3. Measurement system

Since 1997, Reseaux IP Europeen Network Coordination Center (RIPE NCC) has been operating a system called Test Traffic Measurements (TTM) [1] for measuring *OWD*. TTM is an active measurement system, which has implemented the IPPM one-way delay [3] and one-way loss [4] metrics to perform independent measurements of connectivity parameters in the Internet. In TTM, active probe packets containing timestamps are sent from a dedicated measurement PC running FreeBSD operating system on the source network to a similar PC on the destination network. The TTM system is illustrated in Fig. 2. A measurement host is connected 0 hop away from (or, if that is not feasible, as close as possible to) the border router of each participating site. By connecting the host 0 hop away from the border router, we exclude effects of the internal network from our measurements. Each delay measurement is accompanied by a determination of the path between the two locations using a tool like *traceroute*. A *traceroute* between each pair of machines is done approximately 10 times an hour or on average every 6 minutes.

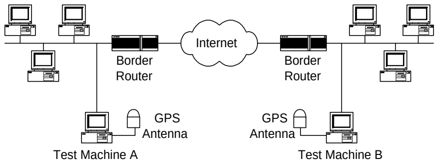

Figure 2. Experimental network environment

The Global Positioning System (GPS), a satellite navigation system developed by the US Department of Defense, is used to avoid clock difference and clock skew between source and destination. The GPS system consists of 24 satellites that continuously broadcast a time- and position-signal. From these signals, a receiver can calculate its position and extract a clock-signal with an accuracy of about 100 ns. GPS also has the additional advantages that the signals are available everywhere on the earth, and that the receivers are relatively cheap and can automatically run without operator control. The GPS antenna used in our work is a Trimble Acutime 2000 [2] (Fig. 3). It generates a pulse-per-second (PPS) output synchronized to Coordinated Universal Time (UTC) within 50 ns (one sigma), outputs a timing packet for each pulse. The value of the clock counter in our machine will be synchronized to the time from the GPS system with an accuracy of several μs. Without GPS system, the previous works [16], [10] have accuracy only in the scale of ms.

Figure 3. Acutime 2000 (GPS receiver)

We have joined with RIPE's TTM project and perform delay and loss measurement from a host at Sezaki laboratory (Tokyo, Japan) to the other hosts in Europe (Netherlands, Sweden, Germany, Denmark, United Kingdom, France, Belgium, Austria, *etc.*) and USA (New York, California, Texas, Denver, Colorado, *etc.*). The probe packets are 128 bytes long, and contain a UDP frame with destination port 6000, and a UDP payload of 100 bytes (header length is 28 bytes). This is the TTM system Type-P definition for the Type-P-One-way-Delay metric framework.

The transmission interval between two consecutive packets has been randomized according to a Poisson distribution. This prevents the synchronization of the test-traffic with other events on the Internet. The Poisson scheduling process is run once for each pair of machines for the entire measurement interval. The receiving program is based on the Berkeley Packet Filter software (BPF). BPF provides a raw interface to the data-link layer and can access the packets as soon as they are released to the O/S by the Ethernet interface.

4. Bottleneck bandwidth estimation

4.1. Methodology

Based on timestamp field, *OWD* is measured by subtracting arrival time from transmission time. After that we consider network characteristics from measured delay, loss and *traceroute*'s data. Next bottleneck bandwidth is estimated by using *EBBP*, an algorithm based on Bolot [6], [5] and Huang [10]. According to equation 3, small value of δ (20, 50, and 100 ms) is used for heavy load situation in which packet-pair phenomenon occurs. We make *OWD* phase plot graph as shown in Fig. 11 ($OWD_{n+1}\ vs\ OWD_n$), and try to find the intercept ($\delta - s/B_b$). Because we know s and δ, therefore we can calculate B_b, bottleneck bandwidth. From derived B_b, we can do reverse calculation, i.e. calculate intercept for other values of transmission interval (δ).

We try to find the intercept from phase plot graph by using the *EBBP* algorithm. The detail of *EBBP* algorithm is as follows.

1. Extract useful samples by choosing packets from percentile 5th to percentile 95th. Packet-pair phenomenon may not occur in the case of low delay because transmission interval is too large. On the other hand, the cause of high delay may be cross traffics between the pair of probe packets.

2. Quantize extracted samples for ease in finding intercept. We use 0.1 ms as quantization level (q). The reason for using 0.1 ms as quantization level is explained in section 5.2. The quantization graph is shown in Fig. 4.

3. Calculate the number of quantized samples for each quantization level (each small square box in Fig. 4). The value of each point $Q(x', y')$ is added by 1, if we find one point within $(x' \pm q/2, y' \pm q/2)$.

$$\begin{aligned} Q(x', y') &= Q(x', y') + 1, \\ \forall P(x, y) &\in (x' \pm q/2, y' \pm q/2) \end{aligned} \qquad (4)$$

4. Because valid samples should be closely clustered around the correct values, while incorrect samples should not be clustered around any one value. Therefore we estimate the intercept by finding a line where many markers concentrate in. We have already known the slope of that line, then we just need to detect the intercept. We use a metric A shown in equation 5.

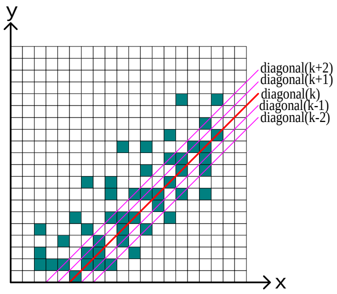

Figure 4. Quantization graph

$$A(k) = \left| \sum_{j=k}^{k+2} diagonal(j) - \sum_{j=k-1}^{k-2} diagonal(j) \right| \qquad (5)$$

$$diagonal(k) = \sum_{j=0}^{end} Q(j+k, j) \qquad (6)$$

The $diagonal(k)$ (equation 6) is the sum of all points in the k^{th} diagonal from main diagonal. By calculating the absolute value of the difference between two sides of $diagonal(k)$, we obtain a metric relative to the intercept k as in equation 5. If A becomes large at k, it means the difference of number of markers between upper-left side and lower-right side of the graph is large. It is a possible intercept candidate. Therefore, the intercept should be k, when A achieves maximum (Fig. 5).

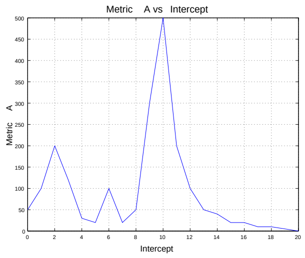

Figure 5. Metric A vs x-coordinate intercept

5. Rewrite Bolot's equation (equation 3) by substituing *OWD* for *RTT* (equation 7).

$$owd_{n+1} = owd_n - (\delta - s/B_b) \tag{7}$$

Thus, bottleneck bandwidth can be estimated from derived intercept by using equation 8.

$$\begin{aligned} k &= \delta - s/B_b \\ B_b &= s/(\delta - k) \end{aligned} \tag{8}$$

4.2. Detectable range of bandwidth

The purpose of this section is to answer the following question – *Which transmission rate is the most suitable for bandwidth estimation?* This is an important question for active measurement because if packets are sent with high rate, they will disturb network and estimation will be invalid. In order to answer this question, let's consider equation 7. The magnitude of intercept must be more than zero. Equation can be proved as follows.

$$\begin{aligned} \delta - \frac{s}{B_b} &> 0 \\ \delta &> \frac{s}{B_b} \\ \frac{s}{R} &> \frac{s}{B_b} \\ B_b &> R \end{aligned} \tag{9}$$

Denote average transmission rate as R, which equals to packet size divided by transmission interval. From equation 9, transmission rate should be smaller than bottleneck bandwidth ($R < B_b$). It means that the range of detectable bottleneck bandwidth is limited by the shade zone in Fig. 6, i.e. we can detect only bandwidht which is more than transmission rate.

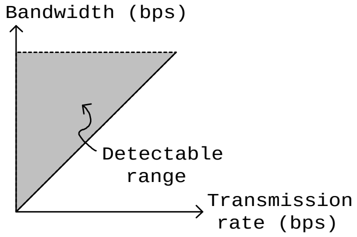

Figure 6. Detectable range of bottleneck bandwidth

Let's consider parameter R in detail. Transmission rate (R) depends on two variables: packet size (s) and transmission interval (δ).

- If value of R is small, the detectable range will be large. This condition can be achieved by using small s and/or large δ. This is desired condition because detectable range is large (i.e. we can detect low bandwidth link) and low rate probe packets do not disturb other traffics in the network. But packet-pair phenomenon may not occur, if probe packet is sent with too low rate.
- If value of R is large, the detectable range will be small. This condition can be achieved by using large s and/or small δ. We can assure that packet-pair phenomenon occurs with high rate probe packets. But this condition is not desirable because we cannot detect low (narrow) bandwidth link.

In order to obtain large detectable range of bottleneck bandwidth, we should decrease transmission rate as low as we assure that packet-pair phenomenon still occurs.

5. Results

5.1. Delay consideration

First, we show measured delay from one pair of participated machines. It is *OWD* measurement from a host in USA to a host in Europe. Figure 7 and 8 show *OWD* vs time (also the number of hops in the same graph) on May 28, 2002 (one-day data) and May 22-28, 2002 (one-week data) respectively. *OWD* is smooth in Fig. 7 except loss between 10:00-11:00. Regarding one-week data in Fig. 8, change in delay came out on the night of May 24^{th}, the number of hops also changes in that time. Change in the routing vectors can often be used to explain why the median delay between two points suddenly changes.

Figure 9 is *OWD* measurement from a host in Europe to a host in our laboratory (Tokyo, Japan) on September 20, 2002. From both left and right graph, *OWD* can be distinguished into three different values. Rerouting is one of the reasons of changes in delay. Therefore, we can use *OWD* in detecting path rerouting.

The next result is *OWD* measurement between two hosts at the same time. *OWD* was measured between a host in our laboratory (Tokyo, Japan) and a host in Europe on August 23, 2002. *OWD* was compared for each direction: uplink and downlink (Fig. 10). On August 23^{rd}, the number of hops for uplink path is 28 hops, while the number of hops for downlink path is 25 hops. Different number of hops refers to different route (average *OWD* for each direction is different as well); hence, this route is asymmetric route. This result aligns with *traceroute*'s data. We can prove that all of Internet path is not symmetrical route as widely believed. Asymmetrical route is one of the problems for *RTT* measurement.

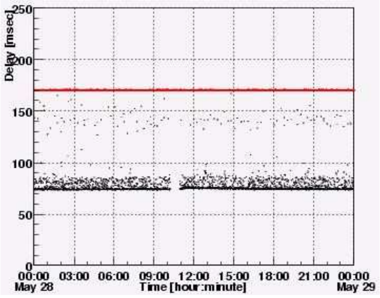

Figure 7. OWD vs time on May 28, 2002 (the upper line is the number of hops×10)

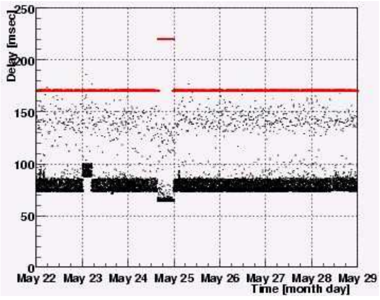

Figure 8. OWD vs time on May 22-28, 2002 (the upper line is the number of hops×10)

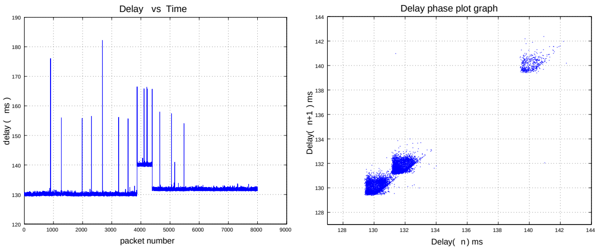

Figure 9. OWD on September 20, 2000: (left) OWD vs packet number; (right) OWD phase plot graph

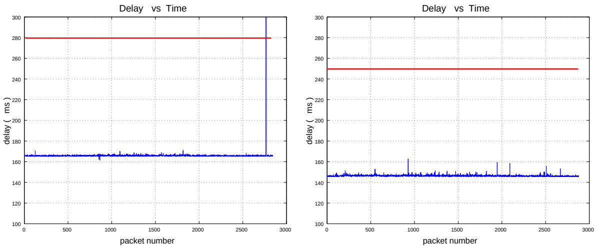

Figure 10. OWD vs time (the upper line is the number of hops×10): (left) uplink path; (right) downlink path

5.2. Bottleneck bandwidth estimation

We performed *OWD* measurement between a host in our laboratory (Tokyo, Japan) and a host in Europe to estimate bottleneck bandwidth by using fixed size probe packet (s = 128 bytes); therefore, transmission rate depends only on transmission interval. Three measurements were held on September 20, 2002 in accordance with three average transmission intervals (δ): 20, 50, and 100 ms, in other words, three transmission rates (R): 50, 20, and 10 $kbps$ respectively. Henceforth, we will mention transmission rate instead of transmission interval. 10-$kbps$, 20-$kbps$, and 50-$kbps$ rates were sent at 14:00–15:00 GMT, 10:00–12:00 GMT, and 00:00–01:00 GMT respectively. *OWD* phase plot graph for each transmission rate is shown in Fig. 11, Fig. 12, and Fig. 13.

The results of bottleneck bandwidth estimation are shown in table 2. The first column is characteristics for each measurement. R refers to transmission rate; time refers to the range of time that the experiment was done; N refers to number of probe packets; δ refers to average transmission interval; $\overline{owd}$ refers to average one-way delay; k refers to intercept of phase plot graph; and B_b refers to estimated bottleneck bandwidth. 10-$kbps$, 20-$kbps$, and 50-$kbps$ rates are approximate value, the exact value is shown at the first row of the table. Since the samples of 50-$kbps$ rate are much more than other measurements (over 160,000 samples), data is divided equally into two groups (30 minutes each group) in order to reduce analysis time. The others (10-$kbps$ and 20-$kbps$ rate) are one-hour measurement. Although we measure the same route, each bottleneck bandwidth estimation yields unequal rate. As mentioned in section 4.2, the lower the transmission rate, the lower the bandwidth is detectable. In addition, detectable bandwidth is always more than transmission rate. All of estimated values in table 2 are greater than transmission rate insignificantly, this implies that this route has very low bottleneck bandwidth.

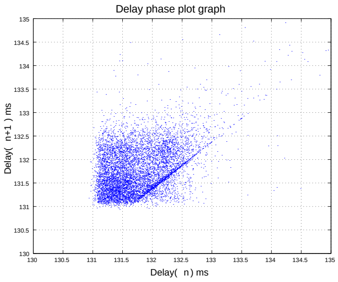

Figure 11. OWD phase plot graph for 10-kbps transmission rate

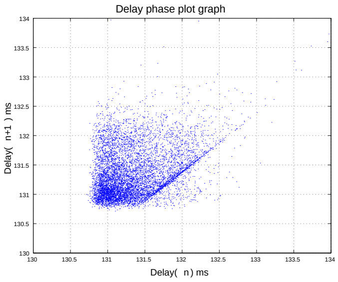

Figure 12. OWD phase plot graph for 20-kbps transmission rate

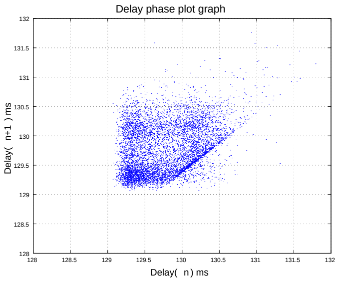

Figure 13. OWD phase plot graph for 50-kbps transmission rate

Table 2. Bottleneck bandwidth estimation

R ($kbps$)	10.17	19.72	47.56	49.21
perios (hours)	1	1	0.5	0.5
N (packets)	35,348	68,531	83,376	83,376
δ (ms)	100.65	51.92	21.53	20.81
$\overline{owd}$ (ms)	131.70	131.51	131.20	131.20
k (ms)	–0.1	–0.1	–0.2	–0.2
B_b ($kbps$)	10.18	19.76	48.01	49.68

Table 3 shows bottleneck bandwidth estimation for 20-$kbps$ rate by varying the period of measurement. We analyze 2-hour (column 2), 1-hour (column 3,4), and 30-minute (column 5) data. All of the periods yield the same result. Thus, dividing data into small groups can reduce cost of time in analysis, while it does not have significant effect on the estimated value.

Table 3. Bottleneck bandwidth estimation: varying the period of measurement

period (hours)	2	1	1	0.5
R ($kbps$)	19.16	19.72	19.72	19.72
N (packets)	137,062	68,531	68,531	34,575
δ (ms)	52.22	51.92	51.92	51.92
$\overline{owd}$ (ms)	131.55	131.51	131.59	131.47
k (ms)	–0.1	–0.1	–0.1	–0.1
B_b ($kbps$)	19.65	19.76	19.76	19.76

In section 4.1, 0.1 ms is chosen as quantization level (q). Here, we change q to be 0.05 ms and analyze the data again. Both 0.1 and 0.05 ms give the same result (see Table 4). Of course, 0.05 ms level uses more analysis time than 0.1 ms level. 0.1 ms is small enough to give accurate result. Therefore, 0.1 ms is set as quantization level in our *EBBP* algorithm.

Table 4. Bottleneck bandwidth estimation: varying quantization level

q (ms)	0.1	0.05	0.1	0.05
R ($kbps$)	10.17	10.17	19.72	19.72
perios (hours)	1	1	0.5	0.5
N (packets)	35,348	35,348	34,575	34,575
δ (ms)	100.65	100.65	51.92	51.92
$\overline{owd}$ (ms)	131.70	131.70	131.47	131.47
k (ms)	–0.1	–0.1	–0.1	–0.1
B_b ($kbps$)	10.18	10.18	19.76	19.76

6. Conclusion

This paper discussed network measurement techniques emphasized on packet-pair model. There are many methods in measuring traffic: active/passive measurement, round-trip/one-way measurement. Each method has its own advantages and disadvantages. The best method depends on measuring metrics, network environments, application tools, and several conditions in measurement. We used one-way delay and loss measurement with GPS's time synchronization, which yielded adequate precision in measuring *OWD*. We proposed *Estimating Bottleneck Bandwidth using Packet-pair (EBBP)* algorithm, which based on Bolot's and Huang's algorithm and discussed how to choose transmission rate and detectable range of bottleneck bandwidth. *EBBP* algorithm has limit in detectable range of bandwidth according to transmission rate. Transmission rate can be changed to receive desired detectable range. Results of bottleneck bandwidth estimation were shown by varying some parameters, i.e. transmission rate, the period of measurement, and quantization level. After that, we concluded the results of each parameter. Analyzing delay and loss gave many useful informations for general users and network operators such as bottleneck bandwidth, path rerouting, path asymmetry, and so on.

7. Future works

Hitherto, our work has been done by using measurement data from only 4-5 hosts (2-3 pairs) in Japan, Europe, and USA. Whereas, there are approximately 50 nodes participated in RIPE TTM project. Hence we plan to use data from several hosts for analysis by setting a host in our laboratory (Tokyo, Japan) to be a main host and choose partners from other participated hosts both in Europe and USA.

As mentioned above, path rerouting can be detected from changes in delay. We plan to propose path rerouting detection algorithm based on *OWD* variation to study rerouting characteristic in the Internet. In addition, it is believed that path rerouting often occurs in USA; we also plan to investigate the route in USA.

References

[1] RIPE NCC test traffic measurement. http://www.ripe.net/ripencc/mem-services/ttm/.

[2] Trimble acutime 2000. http://www.trimble.com.

[3] G. Almes, S. Kalidindi, and M. Zekauskas. RFC 2679: A one-way delay metric for IPPM, September 1999.

[4] G. Almes, S. Kalidindi, and M. Zekauskas. RFC 2680: A one-way packet loss metric for IPPM, September 1999.

[5] J.-C. Bolot. Characterizing end-to-end packet delay and loss behavior in the Internet. *Journal of High Speed Networks*, 2(3):305–323, December 1993.

[6] J.-C. Bolot. End-to-end packet delay and loss behavior in the Internet. In *Proceedings of ACM SIGCOMM*, pages 289–298, September 1993.

[7] R. L. Carter and M. E. Crovella. Measuring bottleneck link speed in packet-switched networks. *Performance Evaluation*, 27&28:297–318, December 1996.

[8] C. Dovrolis, P. Ramanathan, and D. Moore. What do packet dispersion techniques measure? In *Proceedings of IEEE INFOCOM*, April 2001.

[9] A. B. Downey. Using pathchar to estimate Internet link characteristics. In *Proceeding of ACM SIGCOMM*, pages 241–250, September 1999.

[10] L. Huang. An analysis of end-to-end delay on Internet. Master's thesis, The University of Tokyo, March 2001.

[11] V. Jacobson. pathchar: a tool to infer characteristics of Internet paths, 1997. ftp://ftp.ee.lbl.gov/pathchar/.

[12] M. Jain and C. Dovrolis. Pathload: a measurement tool for end-to-end available bandwidth. In *Proceeding of Passive and Active Measurement Workshop (PAM)*, Colorado, USA, March 2002.

[13] K. Lai and M. Baker. Measuring bandwidth. In *Proceeding of IEEE INFOCOM*, pages 235–245, March 1999.

[14] K. Lai and M. Baker. Measuring link bandwidths using a deterministic model of packet delay. In *Proceeding of ACM SIGCOMM*, pages 283–294, August 2000.

[15] B. A. Mah. pchar: A tool for measuring Internet path characteristics, 1999. http://www.employees.org/b̃mah/Software/pchar/.

[16] V. Paxson. *Measurements and Analysis of End-to-End Internet Dynamics*. PhD thesis, University of California, Berkeley, April 1997.

[17] H. Tangmunarunkit, R. Govindan, S. Jamin, S. Shenker, and W. Willinger. Network topology generators: Degree-based vs. structural. In *Proceeding of ACM SIGCOMM*, August 2002.

[18] Y. Tobe, Y. Tamura, H. Aida, and H. Tokuda. Detection of change in one-way delay for analyzing the path status. In *Proceeding of Passive and Active Measurement Workshop (PAM)*, Hamilton, New Zealand, April 2000.