One-way Delay Measurement and Bottleneck Bandwidth Estimation

Niwat Thepvilojanapong†, Yoshito Tobe‡, and Kaoru Sezaki††

†Graduate School of Information Science and Technology, University of Tokyo
‡Department of Information Systems and Multimedia Designs, Tokyo Denki University
††Center for Spatial Information Science, University of Tokyo

Abstract We have participated in 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 path properties such as congestion between each host. 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*s instead of round trip delay. Every host uses GPS receiver to avoid clock difference problem. 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 network characteristics measurement is important to a variety of network applications. Unfortunately, accurate measurement is difficult because of heterogeneity of today's Internet.

There are many interesting network characteristics such as delay, loss, bandwidth and so on. We first distinguish two different definitions of bandwidth, bottleneck bandwidth and available bandwidth. The bottleneck bandwidth of a route is the ideal bandwidth of the lowest bandwidth link (the bottleneck 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. The bottleneck bandwidth is not affected by other traffic. 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 traffic along that route.

We can divide measurement techniques in 2 main methods, i.e. passive and active methods. Passive method uses existing traffic in the network, on the other hand, active method sends probe packets into 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 [1], [2], [3], [5], [7], [8], estimating available bandwidth [11], and estimating per-link bandwidth, latency, and loss [4], [6], [9], [10].

Understanding these characteristics is important for the proper design of network algorithms such as routing and flow control algorithms, for the dimensioning of buffers and link capacity, for choosing parameters in simulation and analytic studies, and for proving and improving network topology generators [16]. 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.

We perform active measurement to measure one-way delay (*OWD*) 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 congestion between each host. 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 [1], but we use *OWD*s instead of round trip delays. Previous works [5], [8] 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 delay, extract useful samples, quantize extracted samples, and find intercept of phase plot graph by *EBBP*. Finally, we can estimate bottleneck bandwidth of the path from know intercept and Bolot's eqution.

In fact, there may be more than one bottleneck link along the path, e.g. in rerouted path[1] and in long

---

[1] route changing because of any reason such as link failure, congestion avoidance, *etc.*

path[2]. The former is natural that bottleneck link will change, if the path changes (as we have already expressed in the definition of the bottleneck bandwidth). The latter seems to conflict our bottleneck bandwidth definition that the bottleneck link should be unique along any path. In this case, we mean the links at which congestion often occurs but they are not the narrowest link along the path. We call these links, frequent congested links, and call bandwidth of these links, *congested link bandwidth.* Our proposed *EBBP* can estimate this *congested link bandwidth.*

The remainder of the paper is organized as follows. Section 2 reviews basis in packet-pair model, also including techniques in estimating bottleneck bandwidth. Section 3 expresses experimental environment and methodology. Section 4 shows results from our experiment. Finally, we conclude this paper in section 5.

2 Packet-pair model

Many works use packet-pair model to estimate bottleneck bandwidth as proposed in [1], [2], [3], [5], [7], [8]. Packet-pair 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 ($t_i^1 - t_i^0 < s/B_b$; variables' 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. Therefore, we can estimate bottleneck bandwidth by using equation 1.

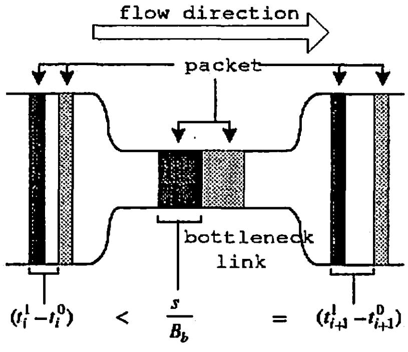

Figure 1: Packet-pair model.

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

Table 1: Variables' 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 (second)
δ	constant transmission interval (packets/second)

The packet-pair model assumes 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 it has over the other techniques are that it measures the true bandwidth of the network (unlike throughput[3]), 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*). On the other hand, there are some of the key problems with current packet-pair algorithm: queuing failure, competing traffic, probe packet drops, and downstream congestion. We can use filtering method to deal with these such problems.

2.1 Bolot's technique

Bolot [1], [2] used 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 analyze the end-to-end packet delay and loss behavior in the Internet. He let the source host be the same as the destination host to avoid clock difference between two machines. Then he compared the *RTT*s of adjacent packets from a sequence sent at regular intervals. The *RTT*s of successive packets were plotted against each other (t_n^k *vs* t_n^{k-1}), called *phase plot graph.* After that he analyzed measured delays and estimated bottleneck bandwidth.

He used large value of δ 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, he used small value of δ for heavy load situation (equation 3). Then he estimated bottleneck bandwidth from calculated intercept ($\delta - s/B_b$) of phase plot graph.

$$\text{Light load:} \quad rtt_{n+1} = rtt_n + \epsilon_n \tag{2}$$

$$\text{Heavy load:} \quad rtt_{n+1} = rtt_n - (\delta - s/B_b) \tag{3}$$

---

[2] We refer long path to the number of hops that the packet traverses.

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

2.2 Huang's technique

Huang's algorithm [8] was mainly based on Bolot. He proposed technique to find the intercept from phase plot graph. Algorithm is as follows: (1) extracts packets at the upper-right part of the graph by using twice of the minimum delay as threshold in extraction; (2) quantizes extracted samples; (3) estimates the intercept by finding a line where many markers concentrate in. Because he used OWDs instead of RTTs, thus the possibility of cross traffics decreases.

3 Experiment

3.1 Experimental environment

Since 1997, RIPE NCC (Reseaux IP Europeen Network Coordination Center) has been operating a system called Test Traffic Measurements (TTM) [12] for measuring OWD. TTM is an active measurement system, which has implemented the IPPM one-way delay [14] and one-way loss [15] metrics to perform independent measurements of connectivity parameters in the Internet. In TTM, active probe packets containing time-stamps are sent from a dedicated measurement PC running FreeBSD 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 hops 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.

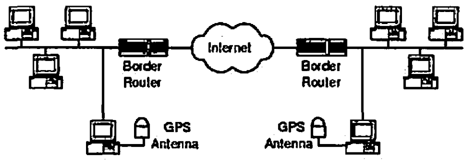

Figure 2: Experimental network environment.

To avoid clock difference and clock skew between source and destination, we use the Global Positioning System (GPS), a satellite navigation system developed by the US Department of Defense. The GPS system consisted 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 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 that we use is a Trimble Acutime 2000 [17] (Fig. 3). It generates a pulse-per-second (PPS) output synchronized to UTC (Coordinated Universal Time) within 50 ns (one sigma), outputting 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 [5], [8] have accuracy only in the scale of ms.

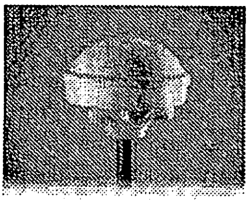

Figure 3: Acutime 2000 (GPS receiver).

We have joined with RIPE's TTM project and perform delay and loss measurement from a host at Sezaki laboratory to other hosts in Europe (Netherlands, Sweden, Germany, Denmark, United Kingdom, France, New Zealand, *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. This is the TTM systems Type-P definition for the Type-P-One-way-Delay metric framework.

3.2 Methodology

Based on timestamp field, we measure OWDs by subtracting arrival time from transmission time. Then we consider network characteristics from measured delay, loss and *traceroute*'s data. Next we estimate bottleneck bandwidth by using an algorithm based on Bolot [1], [2] and Huang [8]. We call our proposed algorithm, *EBBP*. From equation 3, we use small value of δ for heavy load situation in which packet-pair phenomenon occurs. We make OWD phase plot graph as shown in Fig. 4 (OWD_{n+1} *vs* OWD_n), and try to find the intercept ($\delta - s/B_b$). Fortunately, we knew s and δ, therefore we can calculate B_b which is the bottleneck bandwidth. From known B_b, we can calculate intercept for other values of transmission interval (δ).

We try to find the intercept form phase plot graph by using the following algorithm.

1. Extract useful samples by choosing packets from percentile 5^{th} to percentile 95^{th}. Packet-pair phenomenon may not occur in the case of low delay because transmission interval is too large. On the other hand, high delay means there are cross traffics between the pair of probe packets.
2. Quantize extracted samples for ease in finding intercept. We use 0.1 ms as quantization scale. The quantization graph is shown in Fig. 5.

3. Calculate the number of quantized samples for each quantization level (each small square box in Fig. 5). The value of each point $Q(x', y')$ is added by 1, if we find one point within $(x' \pm 0.05, y' \pm 0.05)$.

$$\begin{aligned} Q(x',y') &= Q(x',y') + 1, \\ \forall P(x,y) &\in (x' \pm 0.05, y' \pm 0.05) \end{aligned} \tag{4}$$

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 as the following formula.

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

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

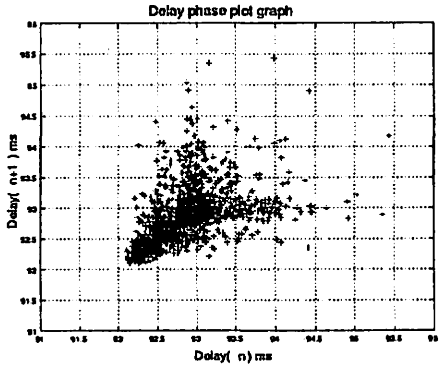

Figure 4: Delay phase plot graph.

The $diagonal(k)$ is the sum of all points in the k^{th} diagonal from main diagonal (equation 6). 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 down-right side of the graph is large. It is a possible intercept candidate. Therefore, the intercept should be k, when A achieves maximum (Fig. 6). Next, we rewrite Bolot's equation (equation 3) as in equation 7. Thus we can estimate bottleneck bandwidth from equation 8.

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

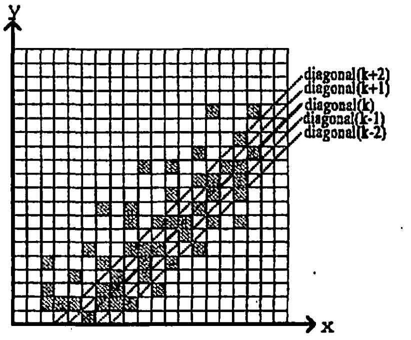

Figure 5: Quantization graph.

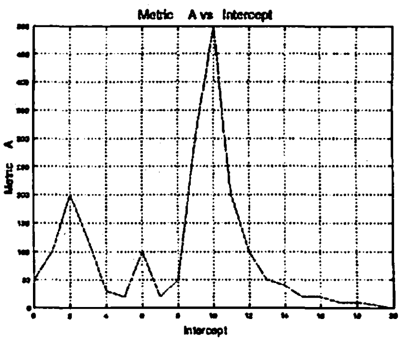

Figure 6: Metric A vs x-coordinate intercept.

$$k = \delta - s/B_b \tag{8}$$

3.3 Refinement

If we use intercept k when A achieves maximum to estimate bottleneck bandwidth, we will not obtain accurate bottleneck bandwidth. We occasionally obtain bandwidth of the most frequent congested link, instead of narrowest bandwidth (bottleneck bandwidth) which we desire to know. Let B_c denote the most frequent congested link bandwidth. Fig. 7 describes how this situation occurs. If congestion occurs at the link (link B_c in Fig. 7) after packet-pair passed bottleneck link (link B_b in Fig. 7), the first packet will be halted from transmission because of queuing. Therefore the second packet has time to pursue the first packet and the gap between two packets will smaller than before. To deal with this problem, we use cumulative distribution function (*cdf*) of the inverse of bandwidth ($1/B$) to improve results from estimating by $EBBP$. Let n be the number of possible intercept k and $i = 0, 1, 2, \ldots, n$. We calculate B_i for each possible intercept k, and calculate *cdf* for each $1/B_i$. The bottleneck bandwidth (narrowest bandwidth) should be near $y = x$ diagonal in the phase plot graph because smaller intercept means smaller bottleneck bandwidth according to equation 3. We conjecture that bottleneck bandwidth should be at *cdf* = 0.9 (Fig. 8). In this case, bandwidth estimated by algorithm in section 3.2 will be bandwidth of the most frequent congested link. We call this bandwidth, ***most frequent congested link bandwidth.***

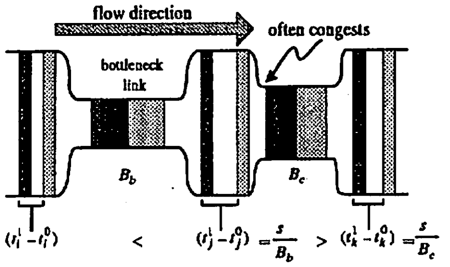

Figure 7: Frequent congested link.

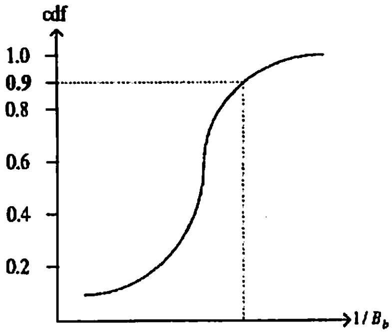

Figure 8: *cdf* of $1/B_b$.

4 Results

4.1 Delay consideration

First we pick up measured delay from one pair of participated machines. Figure 9 and Fig. 10 show delay *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. Delay is smooth in Fig. 9 except loss between 10:00–11:00. When we regard one-week data in Fig. 10, we found change in delay in the night of 24^{th} May, the number of hops also changes in that time. Changes in the routing vectors can often explain why the median delay between two points suddenly changed.

The next one is measured delay between a host in our laboratory and one host in Europe on August 22–24, 2002. We compare one-way delay for each directions: upstream path in Fig. 11 and downstream path in Fig. 12. We can see that on August 24, the number of hops for upstream path is 28, however the number of hops for downstream path is 25. Different path means different route, hence this route is asymmetric. We can prove that all of Internet path is not symmetric route. Asymmetric route is one problem for *RTT* measurement.

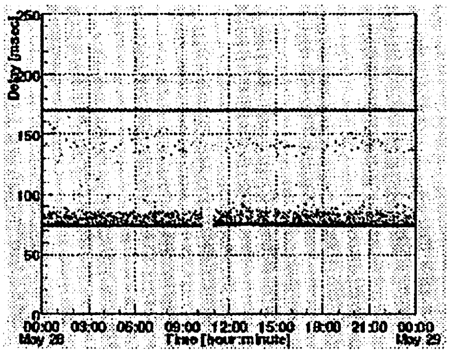

Figure 9: Delay *vs* time on May 28, 2002 (the upper line is the number of hops×10).

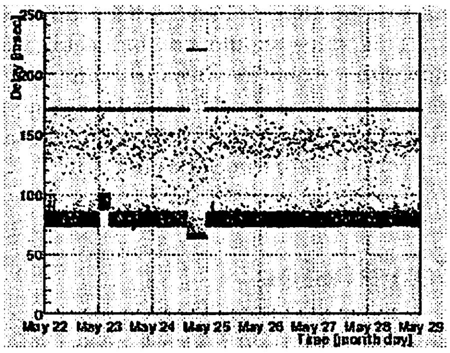

Figure 10: Delay *vs* time on May 22-28, 2002 (the upper line is the number of hops×10).

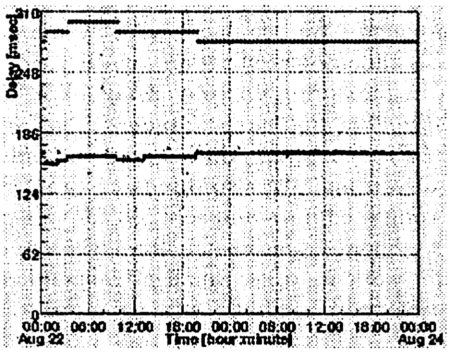

Figure 11: Delay of upstream path (the upper line is the number of hops×10).

4.2 Bottleneck bandwidth estimation

We perform delay measurement from a host in our laboratory to other hosts for estimating bottleneck band-

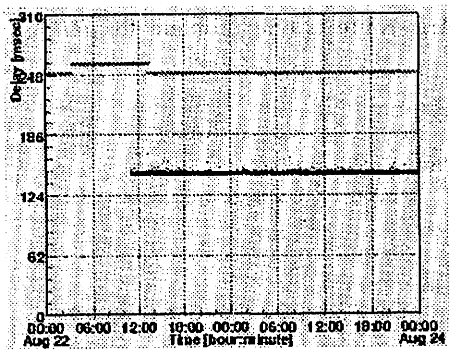

Figure 12: Delay of downstream path (the upper line is the number of hops×10).

width. We use 128-byte packets and transmission interval is 30 second. Using *EBBP*, we obtained intercept = -0.2 ms, and bottleneck bandwidth is approximately 31 $Mbps$. This value should not be accureate bottleneck bandwidth estimation because time interval between packets is too large. If we reduce time interval, we will obtain correct value of intercept and can calculate more accurate bottleneck bandwidth.

5 Conclusion

In this paper, we discussed network measurement techniques concentrating in packet-pair model. There are many methods in measurement: active or passive measurement, round-trip or one-way delay measurement. Each method has advantages and disadvantages. The best method depends on measuring metrics, network environments, application tools, and several conditions in measurement. We use one-way delay measurement with GPS's time synchronization, which yields enough precision in measuring one-way delay. We have proposed *Estimating Bottleneck Bandwidth using Packet-pair (EBBP)* algorithm, which based on Bolot's and Huang's algorithm. We also discussed problem about frequent congested links and solution for this problem. Finally, we showed the results from the experiment. The future work is to use smaller transmission interval to receive more accurate estimation result.

References

[1] Bolot, J.-C.,"End-to-End Packet Delay and Loss Behavior in the Internet," In *Proceeding of ACM SIGCOMM 1993*, pp. 289–298, Sep. 1993.

[2] Bolot, J.-C., "Characterizing End-to-End Packet Delay and Loss Behavior in the Internet," *Journal of High Speed Networks*, Vol. 2, No. 3, pp. 305–323, Dec. 1993.

[3] Lai, K. and Baker, M., "Measuring Bandwidth," In *Proceedings of IEEE INFOCOM'99*, pp. 235–245, Mar. 1999.

[4] Lai, K. and Baker, M., "Measuring Link Bandwidths Using a Deterministic Model of Packet Delay," In *Proceedings of ACM SIGCOMM 2000*, pp. 283–294, Aug. 2000.

[5] Paxson, V., *Measurements and Analysis of End-to-End Internet Dynamics*, PhD Thesis, University of California, Berkeley, Apr. 1997.

[6] Jacobson, V., *pathchar*, ftp://ftp.ee.lbl.gov/pathchar/, 1997.

[7] Carter, R.L. and Crovella, M.E., "Measuring Bottleneck Link Speed in Packet-Switched Networks," *Performance Evaluation*, Vol. 27&28, pp. 297–318, 1996.

[8] Huang, L., *An Analysis of End-to-End Delay on Internet*, Master's thesis, Graduate School of Engineering, University of Tokyo, Mar. 2001.

[9] Mah, B.A., *pchar*, http://www.employees.org/~bmah/Software/pchar/, 1999.

[10] Downey, A.B., "Using pathchar to Estimate Internet Link Characteristics," In *Proceeding of ACM SIGCOMM 1999*, pp. 241–250, Sept. 1999.

[11] Jain, M. and Dovrolis, C., "Pathload: a measurement tool for end-to-end available bandwidth," *Passive and Active Measurement Workshop 2002*, Mar. 2002.

[12] RIPE NCC Test Traffic Measurement, http://www.ripe.net/ripencc/mem-services/ttm/.

[13] Brun, R. and Rademakers, F., "ROOT - An Object Oriented Data Analysis Framework," In *Proceedings of AIHENP'96 Workshop*, 1996.

[14] Almes, G., Kalidindi, S. and Zekauskas, M., "A One-way Delay Metric for IPPM," *RFC 2679*, Sep. 1999.

[15] Almes, G., Kalidindi, S. and Zekauskas, M., "A One-way Packet Loss Metric for IPPM," *RFC 2680*, Sep. 1999.

[16] Tangmunarunkit, H., Govindan, R., Jamin, S., Shenker, S. and Willinger, W. "Network Topology Generators: Degree-Based vs. Structural," In *Proceedings of ACM SIGCOMM 2002*, Aug. 2002.

[17] Trimble Acutime 2000, http://www.trimble.com.