DiffServ 環境における帯域測定ツールの改良と評価

中村 裕一　加藤淳也　後藤滋樹
早稲田大学大学院 理工学研究科 情報科学専攻

Abstract

本論文は DiffServ を用いた通信回線の帯域の新しい測定方法を提案する. 従来から知られている代表的な帯域測定方法に Packet Pair 方式, Packet Train 方式がある. ただし従来の方法は, ルータがパケットを FCFS (First Come First Serve) で扱うことを前提としている. 従来の方式ではボトルネックとなるリンクの物理的な帯域と実効帯域を測定するのに留まっている. その一方で, DiffServ 環境には最低保証された帯域がある. 本論文は, DiffServ において最低保証される帯域を測定可能とするように, 帯域測定方法を改良した. さらに提案する帯域測定方法の評価を行った.

Design and Implementation of a New Measurement Tool for DiffServ Bandwidth

Yuichi NAKAMURA, Jun-ya KATO and Shigeki GOTO
Department of Information and Computer Science, Graduate School of Science and Engineering, Waseda University

Abstract

This paper proposes a new method for estimating the bandwidth of DiffServ networks. There have been two popular ways of bandwidth estimation. One is the *packet pair* method and the other is the *packet train* method. Their main objectives are to estimate the capacity of a path bandwidth, and throughput (i.e., effective bandwidth) which is determined by the bottleneck link along a path. These tools assume that a router handles packets based on the FCFS (First Come, First Serve) principle. On the other hand, they cannot measure the guaranteed bandwidth of DiffServ networks. Our new method can estimate the guaranteed bandwidth. This paper also shows working examples which evaluates the new method.

1 Introduction

Internet backbone bandwidth is growing rapidly because of the growth of the traffic. ISPs (Internet Service Providers) are connected to each other and the scale of networks is getting larger. More clients are connecting to ISPs. The operators of ISPs need bandwidth monitoring tools which give the information beyond their networks. The users of the Internet also want to know whether they get the right bandwidth that they have paid for.

There have been several bandwidth monitoring tools. They assume a router handles packets based on the FCFS (First Come, First Serve) principle. On the other hand, some ISPs began to use DiffServ (Differentiated Service) to provide QoS (Quality of Service) in their networks. DiffServ classifies packets into *classes*. A router deals with classes at different priorities. That is, a router puts an order of priorities among classes. The packets are forwarded according to priority. DiffServ provides the different class of services for Internet traffic. Unfortunately, some prioritizing model cannot be handled properly by the conventional bandwidth estimation methods because they assume that a router forwards packets based on FCFS. This paper proposes an improved method of *packet pair* and *packet train* techniques to estimate the bandwidth controlled by CBQ (Class Based Queuing).

Our new method can estimate three kinds of bandwidth:

1. path bandwidth (i.e. bottleneck bandwidth)
2. effective bandwidth (throughput)
3. guaranteed bandwidth given by CBQ

Section 2 describes earlier well-known method. Namely, the packet pair and packet train techniques. The experiment is shown in Section 3. It shows that old methods cannot estimate the guaranteed bandwidth. Section 4 introduces a new method for the guaranteed bandwidth estimation. Section 5 concludes this paper.

2 Earlier Methods for Bandwidth Estimation

There are two well-known methods in the literature. Section 2.1 introduces *packet pair* method, and section 2.2 explains *packet train* method.

2.1 Packet Pair Method

The packet pair technique estimates the bottleneck bandwidth along a path from a sender to a receiver. The sending machine transmits two packets of the same size sequentially. The receiving machine measures the time difference of the two packets, and estimates the bottleneck

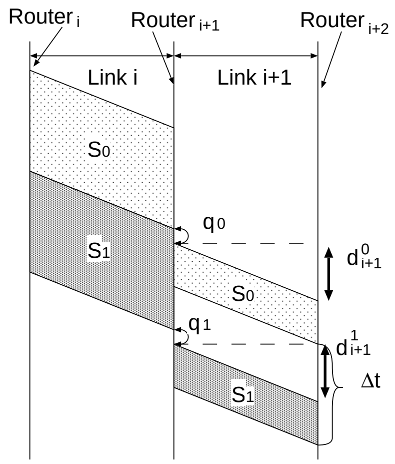

Figure 1: Packet Pair Model in FIFO

bandwidth. It should be noted here that the estimation is based on the assumption that a router handles the queue of packets based on FCFS. The packet pair method is not effective when CBQ is used at the router.

2.1.1 FSFC Forwarding

If a router handles queues of packets based on FCFS, the packet pair technique estimates the bottleneck bandwidth[8]. Figure 1 shows the idea of the packet pair method. The sender transmits two back-to-back packets P_0, P_1 which have the same size ($S_0 = S_1$ (bytes)). Let $Link_i$ be the bottleneck link and its bandwidth is represented by M_i (bps). And let the next link after the bottleneck link be $Link_{i+1}$ and its bandwidth is M_{i+1}.

P_0 takes S_0/M_i (seconds) to travel the bottleneck link. When the second packet P_1 arrives at router $Router_i$, it has to wait in the queue while the first packet P_0 travels the bottleneck link $Link_i$. It takes S_1/M_i (seconds) for P_1 to travel the bottleneck. When the P_1 travels in the bottleneck link, P_0 goes ahead in $Link_{i+1}$ or beyond. That means there is a gap between the first and the second packets as shown in Figure 1. The two packets disperse.

Let the physical delay time (latency) of packet P_0 in link $Link_{i+1}$ be d^0_{i+1} and the physical delay time of P_1 in link $Link_{i+1}$ be d^1_{i+1}. The physical delay has a specific value in each link. We can assume that the physical delay time of P_0 and P_1 are the same. $d^0_{i+1} = d^1_{i+1}$.
Δt means the time difference of two arrival time in Figure 1. It is calculated by the following formula.

$$\frac{S_1}{M_i} + q_1 + d^1_{i+1} + \frac{S_1}{M_{i+1}} = q_0 + d^0_{i+1} + \frac{S_0}{M_{i+1}} + \Delta t$$

Then,

$$\Delta t = \frac{S_1}{M_i} + q_1 + d^1_{i+1} + \frac{S_1}{M_{i+1}} - q_0 - d^0_{i+1} - \frac{S_0}{M_{i+1}}$$

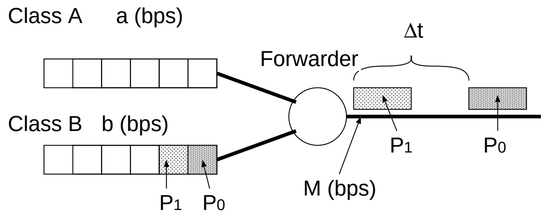

Figure 2: Packet Pair Model in CBQ

If the network is not congested, queuing delays are negligibly small. $q_0 \simeq q_1 = 0$. We have assumed $d^0_{i+1} = d^1_{i+1}$. It is noted that the two packets have the same size, $S_0 = S_1$.

$$\begin{aligned} \Delta t &= \frac{S_1}{M_i} \\ M_i &= \frac{S_1}{\Delta t} \end{aligned} \tag{1}$$

Arrival time dispersion Δt is equal to S_1/M_i (seconds). The packet pair model assumes that the two packets do not have any waiting time except at the bottleneck link.

2.1.2 CBQ Forwarding

CBQ (Class Based Queuing) is a method of realizing DiffServ. CBQ classifies flows into classes and gives guaranteed bandwidth to each class. If one class has less traffic than its guaranteed bandwidth, then another class which has higher traffic than its guaranteed bandwidth can borrow the unused bandwidth of the former class.

If the network uses CBQ, packet pair method estimates the guaranteed bandwidth of the class. Figure 2 shows how the packet pair method estimates the guaranteed bandwidth of a class. Δt is the time dispersion between the arrival time of packets P_0 and P_1.

S_0 is the size of packet P_0 and S_1 is the size of packet P_1. The router (Forwarder) has an outgoing link whose bandwidth is M bps. The guaranteed bandwidth of class A is a (bps) and the given bandwidth of class B is b (bps). The packets, P_0 and P_1, belong to class B. In Δt (seconds), the router transmits $M\Delta t$ (bits) to the outgoing link. The router transmits $a\Delta t$ (bits) that belongs to class A.

$$a\Delta t + S_1 = M\Delta t$$
$$S_1 = (M - a)\Delta t$$
$$\frac{S_1}{\Delta t} = (M - a) = b$$

The above formula shows that the packet pair technique estimates the given bandwidth to the class.

2.2 Packet Train Method

Packet train method is an extension of the packet pair model. It estimates the bandwidth using N back-to-back packets of the same packet size. Figure 3 shows how to estimate the bandwidth using packet train technique. Let

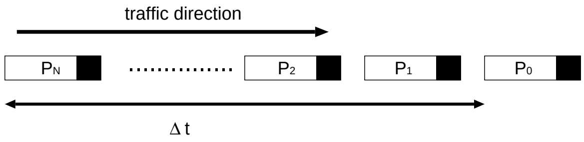

Figure 3: Packet Train Model

Table 2: Traffic Pattern

flow	class	source and destination
packet pair	EF	from H1 to H3
AF traffic(400Kbps)	AF	from H2 to H3
jamming traffic(1.5Mbps)	BE	from H2 to H3

packet size of $P_0 P_1 \cdots P_N$ be all S. The receiver measures the dispersion between the arrival times of P_0 and P_N. The dispersion is Δt (seconds). We estimates the bandwidth b by the following formula:

$$b = \frac{(N-1)S}{\Delta t}$$

The packet train method is more affected by the *cross traffic* than the packet pair model[9]. By this reason, the packet train technique estimates the effective bandwidth of the flow.

3 Experiments by Earlier Methods

We apply the packet pair method and the packet train method to DiffServ networks. We found that the packet pair method estimates the bottleneck bandwidth. And the packet train method estimates the effective bandwidth. However, they can not estimate the guaranteed bandwidth of a class.

The bandwidth of prioritized class depends on the traffic of other classes. If the network is not congested, any class can use the free bandwidth. And we cannot estimate the guaranteed bandwidth. In this experiment, we transmit the fixed rate traffic to make congested networks.

3.1 Experiment Environment

Figure 4 illustrates the experiment environment. There are four routers, R1, R2, R3 and R4. Host H1 sends packet pairs, H2 produces jamming traffic and H3 receives packet pairs and jamming traffic. H1, H2 and H3 are not included in the DiffServ domain. Table 1 shows the specification of the machines. R1 and R4 are the DiffServ Edge Routers and they use ALTQ[16][17]. R1 marks DiffServ Code Point (DSCP) [15] of packets. R1 uses CBQ (Class Based Queuing). R2 and R3 are the DiffServ Core Routers and they guarantee QoS depending on the DSCP. They use Class Based Weighted Fair Queuing[18]. All the links in DiffServ domain are 2Mbps. EF (Explicit Forwarding) class is guaranteed 500Kbps and AF (Assured Forwarding) class is also guaranteed 500Kbps.

Table 1: Network Equipment

Router or Host	Implementation
R1, R4	FreeBSD 4.3 + ALTQ on PC
R2, R3	Cisco 2514 + IOS 12.2(6a) IP Plus
H1, H2, H3	FreeBSD 4.3

3.2 Experiments in Packet Pair Method

Table 2 shows the traffic that we generated. AF traffic and jamming traffic are at the fixed rate. Packet pair is assigned to EF class. AF traffic is at 400Kbps. Jamming traffic is assigned to BE class. H1 sends 100 packet pairs (200 packets in total). The interval of each pair is 50ms. Experiment 1 uses a packet pair of size 500 bytes. Experiment 2 uses a packet pair of size 1450 bytes. Figure 5 shows the result of experiment 1, and Figure 6 shows the histogram of estimated bandwidth. Figure 7 shows the result of experiment 2, and Figure 8 shows the histogram of estimated bandwidth. Using a packet pair of size 500 bytes, we cannot estimate the guaranteed bandwidth of classes. However, it estimates the bottleneck bandwidth of 2Mbps (Figure 5). Using a packet pair of size 1450 bytes, we cannot estimate the guaranteed bandwidth of the class. And it cannot estimate the bottleneck bandwidth either. These errors are due to token bucket meter which will be described in section 3.4.1.

3.3 Experiment in Packet Train Method

The second experiment uses packet train method. We use the same environment as in section 3.2. AF traffic and jamming traffic have the fixed rate. Packet trains are assigned to EF class. AF traffic is at 400Kbps. Jamming traffic is assigned to BE class. We send 100 packet trains (10000 packets in total). The interval of trains is 50ms. Experiment 1 uses 10-packet train of size 500 bytes. Experiment 2 uses 10-packet train of size 1450 bytes. Figure 9 shows the result of experiment 1 and Figure 10 shows the result of experiment 2. Both Figures indicate that packet trains estimate the effective bandwidth.

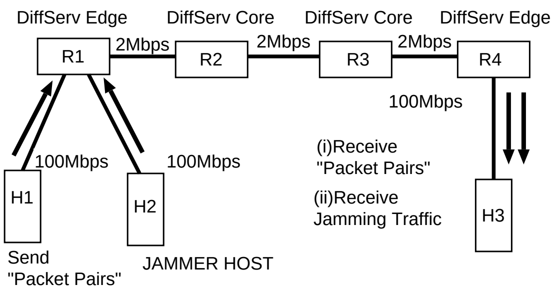

Figure 4: Experiment Network

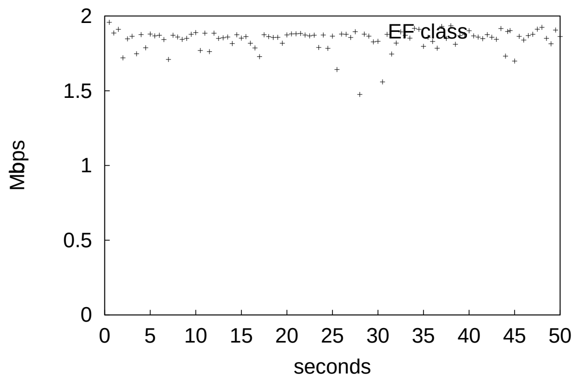

Figure 5: estimation result (bps) , packet size is 500 bytes

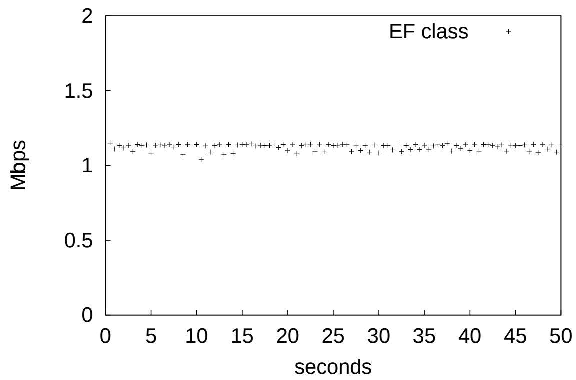

Figure 7: estimation result (bps), packet size is 1450 bytes

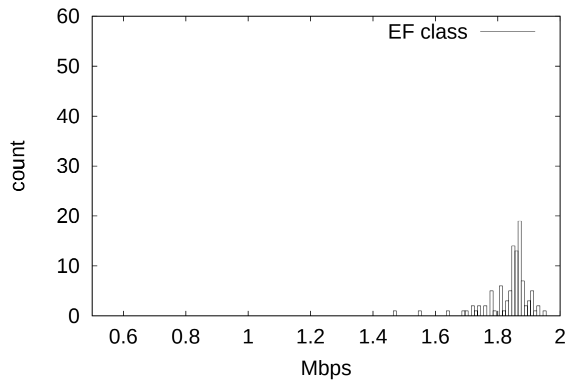

Figure 6: histogram of bps, packet size is 500 bytes

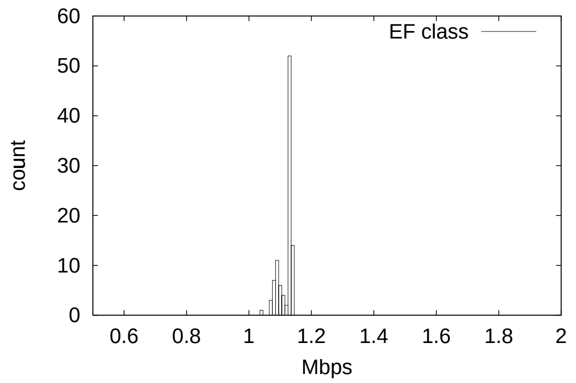

Figure 8: histogram of bps, packet size is 1450 bytes

3.4 Result of Experiments

Experiment results show that we could not estimate the guaranteed bandwidth by earlier methods. This is because CBQ uses token bucket model to regulate the bandwidth.

3.4.1 Token Bucket Model

CBQ is controlled by the token bucket model. The token bucket model determines the maximum burst size and the average data rate using a mathematical model.

The token bucket model is defined as follows. Traffic $A(t)$ is given to a router from 0 (time) to t ($0 \leq t \leq \tau,\ \tau \geq 0$).

The average data rate is ρ and maximum burst size is σ. Token Bucket Model controls $A(t)$ so that the following formula holds:

$$A(t) \leq \sigma + \rho t$$

There is a bucket which can store σ bytes of *tokens* (packets). This bucket stores σ bytes at time 0 and it is full. At the same time, tokens are added to the bucket at the rate of ρ bytes per second. Overflowed tokens are discarded. If a packet of size $L(t)$ arrives at the router at time t, $L(t)$ bytes tokens are removed from the bucket and the packets will be forwarded by the router. If tokens are not enough to exceed the limit of $\sigma + \rho t$, then the packet has to wait until the bucket will get $L(t)$ tokens. As the result, token bucket model can control the average data rate and maximum burst size of the outgoing link.

3.4.2 Packet Train Method

A packet pair of size 500 bytes is forwarded (*bursted*) at the same time because the size is small and the router uses Token Bucket Model. It is the reason why the packet pairs of size 500 bytes estimate the bottleneck bandwidth. On the other hand, a packet pair of size 1450 bytes are not forwarded at the same time, because the size is large. Thus, 1450 bytes packets cannot estimate the guaranteed bandwidth of the class. We will propose a new method to estimate the guaranteed bandwidth in the next section.

4 New Method for Bandwidth Estimation

We propose a newly improved method of packet train model. The conventional packet train method only watches time dispersion between the arrival time of the first packet and the last packet (Figure 3). If CBQ uses the token bucket model and some packets in the train are forwarded (bursted) at the same time, we can't estimate

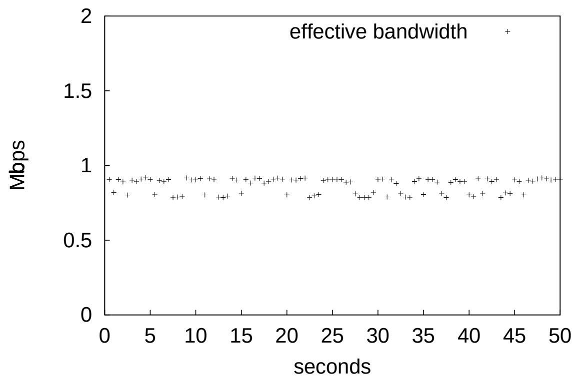

Figure 9: bandwidth estimation (bps), packet size is 500 bytes

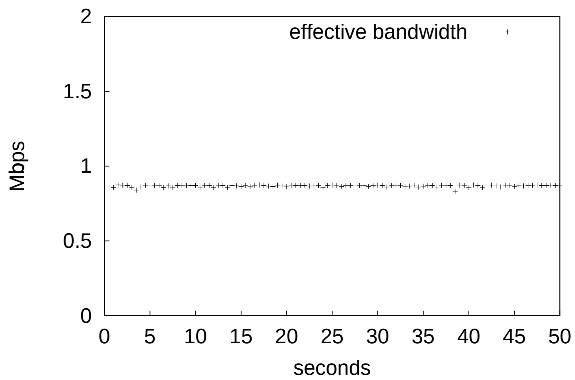

Figure 10: bandwidth estimation (bps), packet size is 1450 bytes

the guaranteed bandwidth by the time dispersion between the first and the last packet.

If some packets in a packet train are bursted and the tokens are not sufficient, the next packet has to wait in a queue. Let the i-th packet is the last packet that is bursted and the tokens are less than the control line of $\sigma + \rho t$.

The size of the $(i + 1)$-th packet is S_{i+1}. The $(i + 1)$-th packet has to wait until enough tokens are added. If the $(i+1)$-th packet is bursted after Δt_w (seconds), the formula below holds:

$$\frac{S_{i+1}}{\Delta t_w} = \rho$$

Figure 13 describes this condition. We can estimate the guaranteed bandwidth using Δt_w. We propose a newly improved packet train method to estimate bandwidth using each dispersion of packets in a train.

We can estimate the bottleneck bandwidth using the smaller packet size based on the result in section 3.2. Figure 9 and Figure 10 indicate that the effective bandwidth can be estimated by packet trains regardless of the packet size. Smaller packet trains are useful to estimating the bottleneck bandwidth, effective bandwidth and the guaranteed bandwidth at the same time.

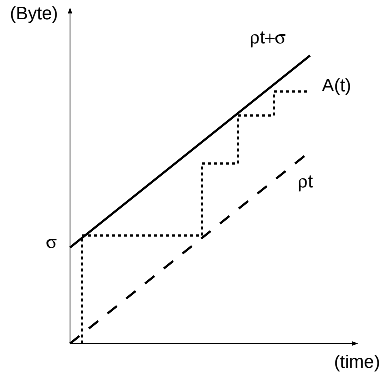

Figure 11: token bucket model

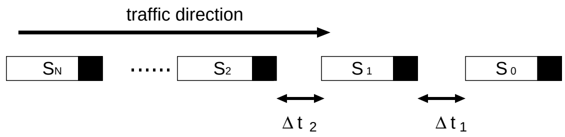

Figure 12: proposed method

4.1 Experiment in the New Method

The experiment result using proposed method are shown below.

4.1.1 Experiment

We evaluate our new method using the same environment as the experiments in section 3.1. We generate various patterns of traffic shown in Table 3. The rate of AF traffic and the rate of jamming traffic are the same as in section 3.1. The packet size of the jamming traffic affects the measurement result. If an EF packet should be forwarded immediately after a BE packets is serviced, the EF packets have to wait until the BE packet is forwarded completely. Dispersion of EF packets may differ depending on a size of packets belonging to the BE class. The experiment is conducted with jamming packet of size 500 (bytes) and 1450 (bytes). The result of experiment No. 1 is shown in Figure 14, the result of experiment No. 2 is in Figure15

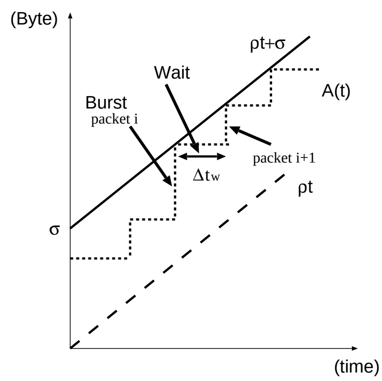

Figure 13: Packet Train Model in Token Bucket Model

Table 3: Packet Size of Traffic Pattern

trial	packet train bytes	jamming traffic bytes
1	1500	1500
2	500	1500
3	500	500

and the result of experiment No. 3 is in Figure 16. In each graph, the bandwidth estimation is performed with 9 dispersions of 10-packet trains.

4.1.2 Results

We discover the following facts:

1. Figure14 shows the bandwidth from 500Kbps to 1Mbps.
2. Figure15 shows the bandwidth from 500Kbps to 2Mbps.
3. Figure16 shows the bandwidth from 500Kbps to 2Mbps.

In section 3.2, we mentioned that the larger packets cannot estimate the bottleneck bandwidth because they are not bursted at the same time. The result of experiment No.1 shows the same result as in section 3.2

A train of smaller size packets (500bytes) can estimate the bottleneck bandwidth. And at the same time, they can estimate the guaranteed 500Kbps.

Thus, a train of small size packets can estimate the bottleneck bandwidth, effective bandwidth and the guaranteed bandwidth at the same time.

5 Conclusion

This paper improves the packet train method to cover the DiffServ network. Using each dispersion time of a packet train, one can estimate the path bandwidth (bottleneck bandwidth), the effective bandwidth and the guaranteed bandwidth. To estimate them, the size of a packet train should be small.

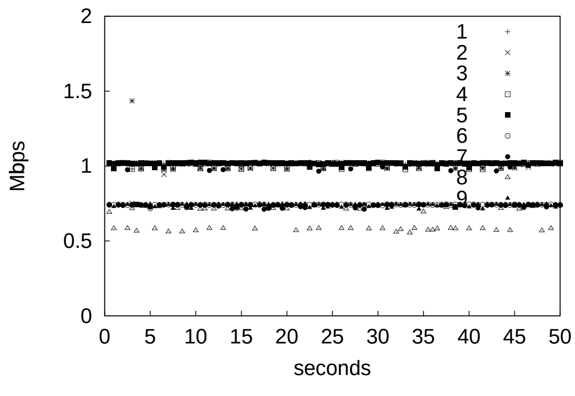

Figure 14: No1. BE 1500bytes, packet train 1500bytes

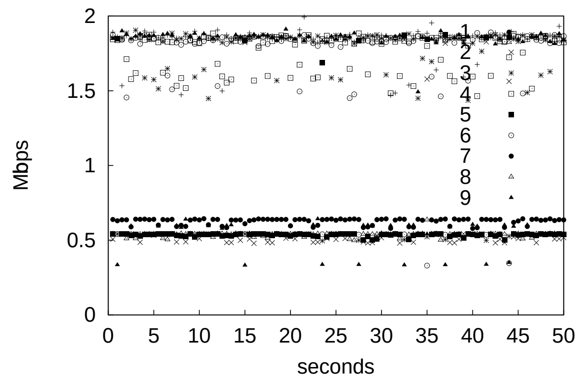

Figure 15: No2. BE1500bytes, packet train 500bytes

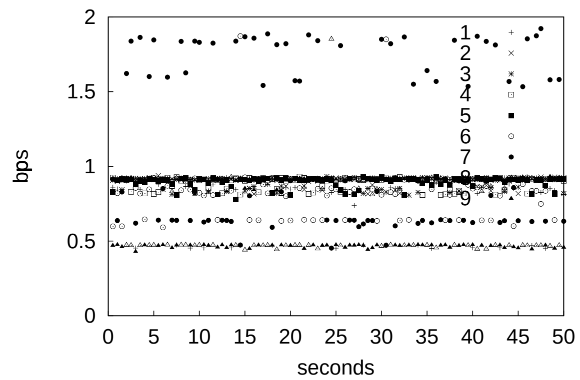

Figure 16: No3. BE500bytes, packet train 500bytes

References

[1] Jean C. Bolot, “End-to-End Packet Delay and Loss Behavior in the Internet,” Proceedings of ACM SIGCOMM, 1994.

[2] Kevin Lai and Mary Baker, “Nettimer: A Tool for Measuring Bottleneck Link Bandwidth,” Proceedings of the USENIX Symposium on Internet Technologies and Systems, 2001.

[3] Kevin Lai and Mary Baker, “Measuring Link Bandwidths Using a Deterministic Model of Packet Delay,” Proceedings of ACM SIGCOMM 2000.

[4] Kevin Lai and Mary Baker, “Measuring Bandwidth,” Proceedings of IEEE INFOCOM ’99, 1999.

[5] Van Jacobson, “pathchar: a tool to infer characteristics of Internet paths,” ftp://ftp.ee.lbl.gov/pathchar/msri-talk.pdf.

[6] Bruce A. Mah, “Estimating Bandwidth and Other Network Properties,” Internet Statistics and Metrics Analysis Workshop on Routing and Topology Data Sets: Correlation and Visualization, 2000.

[7] Bruce A. Mah “Pchar: Child of Pathchar,” DOE NGI Testbed Workshop, 1999.

[8] Vern Paxson, “End-to-End Internet Packet Dynamics,” ACM SIGCOMM ’97, 1997.

[9] Constantinos Dovrolis, Parameswaran Ramanathan, Daivd Moore, “Packet Dispersion Techniques and Capacity Estimation,” IEEE/ACM Transactions in Networking, July, 2001.

[10] Constantinos Dovrolis, Parameswaran Ramanathan, Daivd Moore, “What do packet dispersion techniques measure?” Proceedings of Infocom, 2001.

[11] CAIDA, “Bandwidth Estimation Project,” http://www.caida.org/projects/bwest/.

[12] Guojun Jin, George Yang, Brian R. Crowley, Deborah A, Agarwal, “Network Characterization Service (NCS),” LBNL report

#47892, http://www-didc.lbl.gov/papers/NCS.HPDC01.pdf.

[13] Van Jacbson, "Congestion Avoidance and Control," Proceedings of ACM SIGCOMM, 1988.

[14] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, "An Architecture for Differentiated Service," RFC2475, 1998.

[15] K. Nichols, S. Blake, F. Baker, D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers," RFC2474, 1998.

[16] Kenjiro Cho, "A Framework for Alternate Queueing: Towards Traffic Management by PC-UNIX Based Routers," Proceedings of USENIX 1998 Annual Technical Conference, 1998.

[17] Kenjiro Cho "ALTQ: Alternate Queueing for BSD UNIX," http://www.csl.sony.co.jp/~kjc/software.html#ALTQ.

[18] Cisco systems, "Implementing DiffServ for End-to-End Quality of Service Overview," http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fqos_c/fqcprt7/qcfdfsrv.htm.

[19] S. Floyd, V. Jacobson, "Link-sharing and resource management models for Packet networks," IEEE/ACM transaction on Networking vol. 3, no. 4, Aug-95.