Router buffer re-sizing for short-lived TCP flows

Takeshi Tomioka
Graduate School of Law
Osaka University
1-6 Machikaneyama, Toyonaka, Osaka, 560-0043, Japan
Email: q06065h@lawschool.osaka-u.ac.jp

Go Hasegawa, Masayuki Murata
Graduate School of Information Science and Technology
Osaka University
1-32 Yamadaoka, Suita, Osaka, 565-0871, Japan
Email: {hasegawa, murata}@ist.osaka-u.ac.jp

***Abstract*— Traditionally, the size of router buffers is determined by the bandwidth-delay product discipline (*normal discipline*), which is the product of the link bandwidth and average round-trip time (RTT) of flows passing through the router. However, recent research results have revealed that when the number of flows is sufficiently large, the buffer size can be decreased to the bandwidth-delay product divided by the square-root of the number of flows (*sqrtN discipline*), without introducing under-utilization of the link bandwidth. This assertion has been verified mainly for long-lived flows; in contrast, there has not been a thorough verification of short-lived flows, which are the majority of Internet flows. Furthermore, the effects of network parameters, such as the link bandwidth and propagation delay, have not been investigated. In this paper, we compare the performance of the above two disciplines, normal and sqrtN, for sizing the router buffer. We focus on the performance of both long-lived and short-lived TCP connections traversing the router under various network environments. Through extensive simulations, we show that the sqrtN discipline would degrade the TCP performance in terms of the packet loss ratio and file transmission delay, and it may be useful only when the transferring file size is about 50-100 Kbytes or when the propagation delay between the sender and the receiver hosts is significantly small.**

***Index Terms*— router, buffer size, TCP, bottleneck link, access link, bandwidth, bandwidth delay product**

I. Introduction

Today, many applications rely on Transmission Control Protocol (TCP) [1] to avoid and resolve congestion in the Internet. Although other applications utilize User Datagram Protocol (UDP) [2] to control the network congestion on their own, the proportion of UDP traffic is very small compared with TCP traffic on the current Internet [3]. Therefore, evaluating the performance of TCP traffic on a network is very important. TCP performance is largely affected by the round-trip time (RTT) and packet loss ratio of the network path [4, 5]. The output link buffer of almost all of Internet routers deploys the First-In First-Out (FIFO) discipline, and the size of this buffer affects the RTT and packet loss ratio of TCP connections passing through the router. Packets can accumulate at this buffer and cause a queuing delay and delay-variance. It also causes packet losses when packets arrive at a fully-utilized buffer. Therefore, the packet loss ratio can be reduced by utilizing a larger-sized buffer, but this can cause a larger queuing delay since a larger number of packets are accumulated at the buffer.

The size of router buffers is traditionally determined based on a rule-of-thumb attributed to [6]. As stated in this paper, the size of a router's buffer should be more than $B = C \times RTT$: that is, the product of the link bandwidth and the average RTT of flows that pass through the router. This is well known as the bandwidth-delay product discipline (we call this the *normal discipline* in this paper), and many routers are equipped with a buffer whose size is determined by this discipline. This discipline is also described in a recent RFC [7].

However, according to [8], it is difficult to construct a router buffer based on this discipline due to the hardware limitation. Today's backbone networks generally carry more than 10,000 concurrent flows and their link bandwidth is 2.5 Gbps or 10 Gbps [9]. If the average RTT equals 250 msec, a 10 Gb/s router needs 250 msec $\times 10$ Gbps = 2.5 Gbits for its buffer. The size of the largest commercial static RAM (SRAM) chip is currently 72 Mbits, which means many SRAM chips are needed to provide a 2.5 Gbps buffer. This size also results in large overhead in terms of board size, electrical power consumption, and monetary cost. On the other hand, a dynamic RAM (DRAM) chip is available up to 1 Gbps and it has significant advantages in monetary cost and board size. However, DRAM has a random access time of about tens of ns, which is from 5 times to 10 times slower than that of SRAM. Therefore, the problem will become worse as line-rates increase in the future. In addition, the electrical power consumption of DRAM is larger than that of SRAM. In summary, it is extremely difficult to build a router buffer for current and future high-speed networks based on the normal discipline.

One possible solution for this problem is stated in [8]. It is shown that the router buffer size can be reduced to the bandwidth-delay product divided by the square root of the number of flows, N, that is, $B_{\mathrm{s}} = \frac{C \times \overline{RTT}}{\sqrt{N}}$, when there are many flows (500 or more) passing through the link. We call this guideline the *sqrtN discipline*. The authors in [8] assert that this small buffer size is enough to keep the link utilization as well as that in the normal discipline. In [10], the authors state that we need only tens of packets for the router buffer size when the input link bandwidth is far smaller than the output link bandwidth (for example, 10 Mbps input and 1 Gbps output), and/or when using pacing TCP [11] (TCP which prevents the data packet from being sent in bursts).

However, these studies pay attention only to the utilization of the bottleneck link bandwidth as a performance metric

 ISBN # 1-56555-317-9

in the simulations and implementation experiments, and the performance of TCP flows passing through the router is almost ignored. In addition, the network environments in these experiments are quite limited and the effects of various network parameters, such as link bandwidth and propagation delay, have not been investigated. Furthermore, we believe that the conditions stated in [10] cannot be satisfied in future networks: the link bandwidth of the access network is increasing rapidly in recent years, and pacing TCP is not currently widespread and is not easily implemented, especially when the access link bandwidth increases. One possible way to spacing TCP packets is to use traffic shaping mechanism at the network edge. However, most of traffic shaping mechanisms would arrow small burst of packets, so the short packet-burst generated by the slow start phase of a TCP connection can not be spaced effectively.

Therefore, in this paper, we evaluate the effect of the buffer size on the following, in addition to link utilization: the packet loss ratio and queuing delay at the router, and the performance of TCP flows passing through the router. Especially, we focus on the performance of short-lived TCP connections when a small-sized buffer is used at the bottleneck link, since the performance of a short-lived TCP data transfer is affected not only by various factors of the bottleneck link utilization, but also by factors including the RTT, packet loss ratio and available bandwidth. We also investigate the effect of other network parameters such as the propagation delay and physical capacity of the bottleneck link, and derive the parameter range where the sqrtN discipline is effective or ineffective.

As we know, [12] is the only paper which discusses the effect of the router buffer size on the performance of short-lived TCP connections. It revealed that the packet loss ratio becomes larger when we use the smaller-sized buffer recommended in [8], and it sometimes hinders the performance of TCP data transfer. However, the investigation in [12] was done with a fixed network environment, and the authors only considered congested networks with almost 100% link bandwidth utilization. In this paper, on the other hand, we investigate the effects of the network parameters and consider under-utilized networks where the link utilization is far below 100%. We also consider the realistic distribution of the file sizes which TCP connections transmit, unlike the fixed value for transferred file sizes used in [12].

The rest of the paper is organized as follows. Section 2 reviews the two disciplines for determining router buffer size: that is, the normal discipline and the sqrtN discipline. Section 3 describes the network model, parameter setting, and evaluation metric for the simulations. In Section 4, we show extensive simulation results and discuss router buffer sizing. Section 5 concludes the present paper and gives some future areas of study.

II. Guidelines for router buffer sizing

A. Normal (bandwidth-delay product) discipline

The traditional guideline for setting the buffer size based on the bandwidth-delay product is described in [6]. We call this guideline the normal discipline. In what follows, we introduce the fundamental reasons for the normal discipline. For a detailed explanation, please refer to [6].

The changes in the congestion window size of a TCP connection in the congestion avoidance phase can be modeled as additive-increase and multiplicative-decrease (AIMD) in versions of TCP such as Reno [13] and NewReno [14]. Figure 1 presents the typical behavior of a single TCP-Reno flow passing through a single-bottlenecked-router network. The top graph shows the time evolution of the queue length at the bottleneck router buffer, and the bottom graph shows the changes in the congestion window size of the TCP connection, where B_{max} is the buffer capacity. We assume the bottleneck link bandwidth is C. From time t_1, the sender starts filling the buffer until a packet is dropped because of the full buffer (at time t_2). About one RTT later, the sender receives duplicate ACKs. Then, the sender retransmits the lost packet, and halves its window size from W_{max} to $W_{max}/2$ (at time t_3). Before the time t_3, the sender is allowed to have W_{max} outstanding packets. But after time t_3, the sender is only allowed to have $W_{max}/2$ outstanding packets. So, the sender must stop sending packets until it receives $W_{max}/2$ ACK packets. This means the number of packets in the buffer decreases while the sender stops sending packets (from time t_3 to time t_4). After time t_4, the sender increases it's window size, so the number of packets in the buffer again increases after time t_5.

If the buffer goes empty before time t_5 comes, the router cannot send packets onto the bottleneck link at a constant rate, so the link utilization becomes less than 100%. While the router's buffer is not empty, the sender's ACKs arrival rate equals the bottleneck link bandwidth C. So, the sender stops sending packets for $(W_{max}/2)/C$ seconds to wait for $W_{max}/2$ ACK packets. On the other hand, the buffer is emptied after B_{max}/C seconds. Therefore, if B_{max}/C is less than $(W_{max}/2)/C$, the buffer is emptied. That is, the following condition should be satisfied to avoid the buffer becoming empty:

$$B_{max} \geq W_{max}/2 \tag{1}$$

The amount of data packets that exist on the bottleneck link can be shown as $C \times \overline{RTT}$ where $\overline{RTT}$ is the average RTT value of TCP connections passing through the link. If $W_{max}/2$ is larger than $C \times \overline{RTT}$, we can keep the bottleneck link fully utilized. Therefore, we require the following condition to be satisfied:

$$W_{max}/2 = C \times \overline{RTT} \tag{2}$$

Finally, Equations (1) and (2) lead to

$$B_{max} \geq W_{max}/2 = C \times \overline{RTT} \tag{3}$$

In summary, if $B_{max} \geq C \times \overline{RTT}$ is satisfied, the buffer never goes empty, and we can take full advantage of the bottleneck link capacity.

In a backbone network, many TCP flows share the bottleneck link. However, the above discussion still holds true if the flows are synchronized. When N TCP flows exist at the

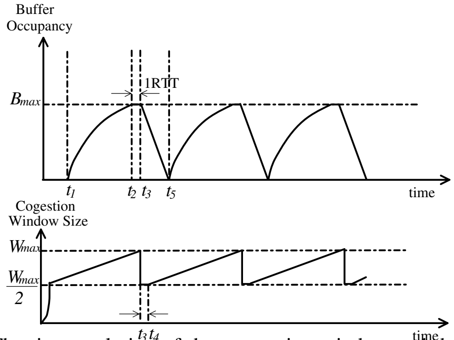


Fig. 1. The time evolution of the congestion window and the queue length

bottleneck link, we can consider that an individual flow has a bandwidth of C/N [15-18]. This means that each flow needs more than $C/N \times \overline{RTT}$ for the buffer capacity. Therefore, the buffer capacity needed can still be shown as follows:

$$(C/N \times \overline{RTT}) \times N = C \times \overline{RTT} \tag{4}$$

B. Square-root discipline

In [8], the authors propose decreasing the router buffer size to the bandwidth-delay product of the network divided by the square-root of N, which is the number of concurrent TCP connections passing through the bottleneck router, when n is sufficiently large (typically larger than 500). In this paper, we call this guideline the *sqrtN discipline*.

The window sizes of TCP connections usually change synchronously when the number of concurrent connections is small and their RTTs are almost equal [15-18]. This means when a certain connection halves its window size, the others do the same simultaneously [15]. This is because of the nature of the drop-tail buffer at the bottleneck link, which causes bursty packet losses when the buffer overflows.

However, flows are not synchronized in many cases. For example, small variations in RTTs or processing times are sufficient to prevent synchronization [19]. The absence of synchronization has been demonstrated in real networks [9, 20]. Even if flows do not have a diversity of RTTs, they can become asynchronous when there are more than 500 concurrent flows [8]. In what follows, we briefly introduce the required buffer size when TCP flows are not synchronized. For details, please refer to [8].

The queue occupancy $Q(t)$ of N flows at time t can be derived by using the congestion window size of each connection $W_i(t)$:

$$Q(t) = max\left(0, \sum_{i=1}^{N} W_i(t) - (\overline{RTT} \times C)\right) \tag{5}$$

Since the average value of the sum of the congestion window size of all flows is obtained as $\overline{W} = \sum_{i=1}^{N} \overline{W_i(t)}$, the average queue occupancy $\overline{Q}$ is given by:

$$\overline{Q} = max(0, \overline{W} - (\overline{RTT} \times C)) \tag{6}$$

$\overline{Q} > 0$, the average congestion window size of each flow, $\overline{W_i}$, can be calculated from Equation (6) as follows:

$$\overline{W_i} = \overline{W}/N = \frac{\overline{RTT} \times C + \overline{Q}}{N} \leq \frac{\overline{RTT} \times C + B_{max}}{N} \tag{7}$$

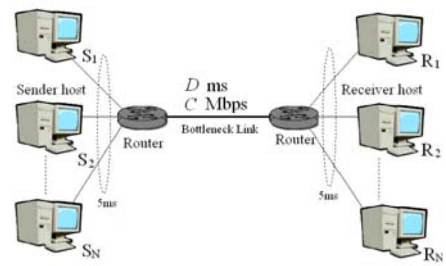


Fig. 2. Network topology for simulations

The standard deviation of distribution of the change in the congestion window size, σ_{w_i}, can be described by the following equation, based on the assumption that the change in the sum of the window size of all connections follows the normal distribution:

$$\sigma_{w_i} = \frac{1}{3\sqrt{3}}\overline{W_i} \tag{8}$$

For a large number of flows, the standard deviation of the sum of the windows, σ_{w_i}, is given by

$$\sigma_w \leq \sqrt{N}\sigma_{w_i} \tag{9}$$

From Equations (7) - (9) the standard deviation of the queue occupancy, Q, is shown as:

$$\sigma_Q = \sigma_w \leq \frac{1}{3\sqrt{3}} \frac{\overline{RTT} \times C + Q}{\sqrt{N}} \leq \frac{\overline{RTT} \times C + B_{max}}{\sqrt{N}} \tag{10}$$

Therefore, we obtain the following lower bound for the link utilization:

$$Util \geq erf\left(\frac{3\sqrt{3}}{2\sqrt{2}} \frac{B_{max}}{\frac{\overline{RTT} \times C + B_{max}}{\sqrt{N}}}\right) \tag{11}$$

For example, if there are 10000 concurrent flows, $Util \geq erf(\frac{3\sqrt{3}}{2\sqrt{2}}) \simeq 0.9899$ when we set $B_{max} = \frac{\overline{RTT} \times C}{\sqrt{N}}$. This result means that we can achieve 98.99% of the link utilization with the buffer whose size is the bandwidth-delay product divided by the square-root of the number of flows, that is, $B_s = \frac{\overline{RTT} \times C}{\sqrt{N}}$. In [8], the effectiveness of the sqrtN discipline is confirmed by the simulations and the experiments, but attention is given mainly to the long-lived TCP flows. For accommodating short-lived flows, it is only stated that small buffers are needed from the aspect of the maintenance of link utilization. It is a straightforward expectation that when we use a smaller buffer, the packet loss ratio increases, which is also shown in [8]. However, there is no description of how the packet loss ratio influences the performance of short-lived traffic.

Then, in the following sections, we clarify the influence of a small size of buffer on short-lived flows through extensive simulations.

III. Evaluation environment

A. Network and traffic model

We evaluate the performance of the two disciplines for buffer sizing using ns-2 [21] simulations. The network model for simulations is depicted in Figure 2. The model consists of sender/receiver terminals (S_1 to S_N and R_1 to R_N), two intermediate routers, and links interconnecting terminals and routers. The link between the two routers is a bottleneck link with a D msec propagation delay and C Mbps bandwidth. The links between the terminals and routers have a 5 msec propagation delay and bandwidth equal to the bottleneck link if not notified. We change N, C, D, and the access link bandwidth in the simulations and investigate the performance of the two buffer sizing disciplines.

We use two kinds of traffic models: P2P traffic and Web traffic. In the P2P traffic model, the sender terminals have an infinite amount of data and continue sending the data by using an FTP-like protocol. In the Web traffic model, on the other hand, the sender terminals determine their data (file) sizes and data transfer intervals based on the Scalable URL Reference Generator (SURGE) model [22]. SURGE is a realistic Web workload generation tool that mimics a set of real users accessing a server.

TABLE I

SUMMARY STATISTICS FOR MODELS USED IN SURGE [22]

Component	Probability Density Function	Parameters
File Sizes - Body	$p(x) = \frac{e^{-(lnx-\mu)^2/2\sigma^2}}{x\sigma\sqrt{2\pi}}$	$\mu = 9.357$ $\sigma = 1.318$
File Sizes - Tail	$p(x) = \alpha k^{\alpha} x^{-\alpha+1}$	$k = 133K$ $\alpha = 1.1$
Inactive OFF Times	$p(x) = \alpha k^{\alpha} x^{-\alpha+1}$	$k = 1$ $\alpha = 1.5$

Table I shows the parameters of the SURGE model. For both traffic models, we change the traffic volume by changing the number of sender/receiver terminals (N).

B. Metrics for the performance evaluation

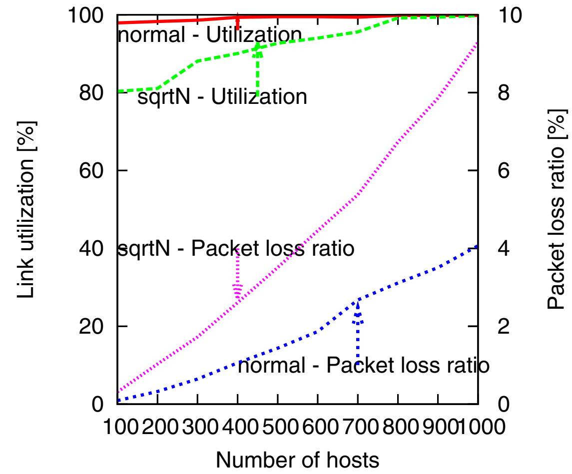
We observe the behavior of the packet at the bottleneck link router. We calculate the link utilization from the number of packets that pass per unit time, and the packet loss ratio from the number of lost packets and the number of packets that arrive at the router. For the Web traffic, we check the file transfer time, which is the time from the beginning of the file transmission to the reception of the ACK packet corresponding to the last packet. The packet loss ratio for each file transfer is also derived to check the relationship between the transferred file size and packet loss ratio.

IV. SIMULATION RESULTS AND DISCUSSIONS

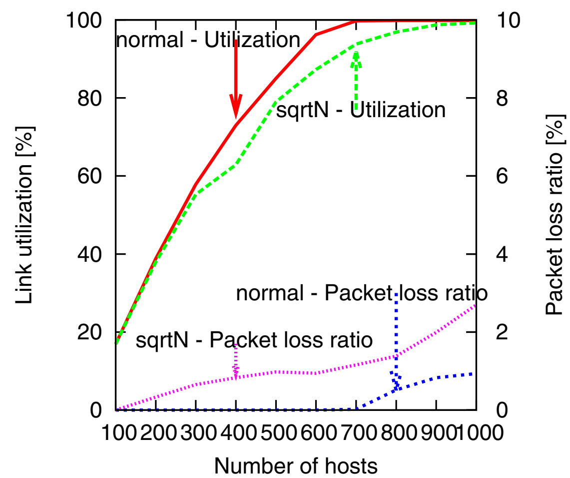
A. Basic performance

Figure 3 shows the change in the link utilization and packet loss ratio when the number of hosts N is changed, where all sender hosts use the P2P traffic model (Figure 3(a)) and the Web traffic model (Figure 3(b)). We set C=100 Mbps and D=90 msec.

Figure 3(a) shows that when the buffer size is determined by the normal discipline, high link utilization can be obtained regardless of the number of hosts. This is simply because the larger buffer size brings the lower packet loss ratio. However,



(a) P2P traffic



(b) Web traffic

Fig. 3. Influence of buffer size to link utilization and packet loss ratio

when the sqrtN discipline is used, the link utilization decreases when the number of hosts becomes small (less than 800 hosts). This degradation is as much as 20% of the bottleneck link bandwidth, especially when the number of multiplexed flows is small. Furthermore, we can recognize the larger packet loss ratio, regardless of the number of flows, compared with the normal discipline. However, we conclude that the assertion in [8] is correct because the link utilization is almost 100% when there is a large enough number of hosts (concurrent flows). That is, when we have sufficiently many co-existing persistent flows, we can reduce the buffer size without degradation of the link utilization.

In Figure 3(b) for Web traffic, we can observe that the link utilization with the sqrtN discipline is also lower than that with the normal discipline when short-lived TCP flows are accommodated. We should note here that the link utilization with the sqrtN discipline becomes degraded in under-utilized networks; even when the link utilization with the normal discipline is around 60-80%, the sqrtN discipline further degrades the link utilization. We also note that the packet loss ratio with the sqrtN discipline does not decrease to zero even when the number of hosts is small, whereas that of the normal discipline becomes zero when the number of hosts is less than 700. This is mainly due to the bursty nature of short-lived Web traffic. That is, the small buffer with the sqrtN discipline cannot absorb the bursts of packets from the short-lived TCP connections in the slow-start phase of their packet transmission. However, the link utilization becomes almost 100% when the number of hosts is sufficiently large. Therefore, the assertion in [8] is also confirmed even with short-lived Web traffic.

In the following, we investigate whether the conclusion in [8] remains true even when the network environment changes, and we check the characteristics of the sqrtN discipline in terms of the performance of each TCP flow passing through the router.

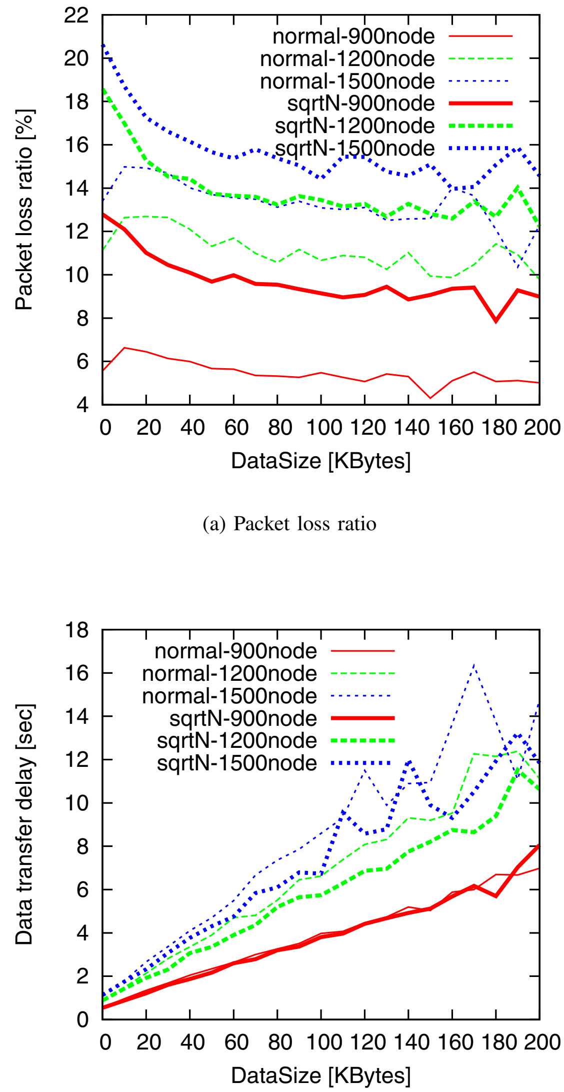
B. *Effect of the change in network environment*

We next discuss whether the normal discipline or the sqrtN discipline should be applied when the network environment changes. This section gives the guidelines for sizing a router buffer for future high-speed networks.

1) Traffic volume: Figures 4 and 5 depict the change of the packet loss ratio and data transfer delay as a function of the transferred file size when the number of hosts, corresponding to the traffic volume, is changed. We utilize the Web traffic model for each sender host, and set C=100 Mbps and D=20 msec (Figure 4) and 90 msec (Figure 5).

Figures 4(a) and 5(a) show that the packet loss ratio with the sqrtN discipline is always higher than that with the normal discipline. This is confirmed by Figure 3 in the previous subsection. We also point out that the packet loss ratio with the sqrtN discipline increases as the transferred file size decreases when D=20 msec, whereas that with the normal discipline remains almost constant. This is because TCP connections with a small data size have a strong bursty nature in their packet transmission, and the smaller buffer with the sqrtN discipline cannot absorb the burstiness.

However, the difference in the packet loss ratio does not affect enough the data transfer delay. From Figures 4(b) and 5(b), we can observe that effect of the high packet loss ratio with the sqrtN discipline to the data transfer delay is small when D=20 msec, whereas it causes a larger transfer delay when D=90 msec. This is because the RTT values of the TCP connections become small when the propagation delay is small, and this feature brings quick detection of the packet losses and their retransmission. Consequently, the sqrtN discipline conceals the bad effect in the increase of packet loss ratio. On the other hand, when the RTTs are large, as in Figure 5, the higher packet loss ratio brings the larger data transfer delay, as we expected.



(a) Packet loss ratio

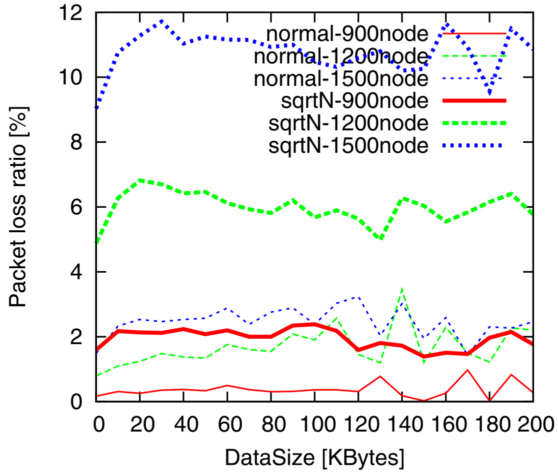
(b) Data transfer delay

Fig. 4. Effect of transferred data size (D=20 msec)

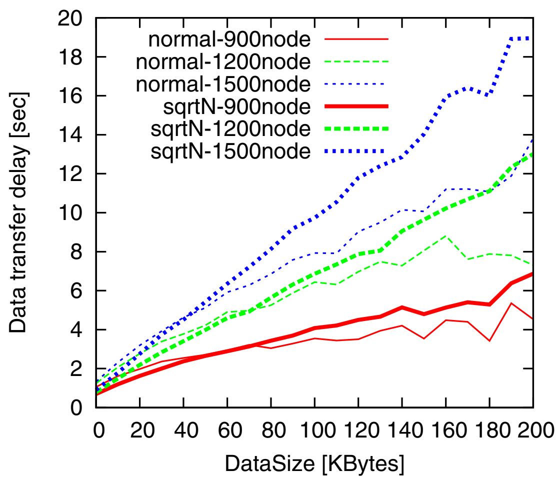
2) Access link bandwidth: Figures 6 and 7 show the change of the packet loss ratio and data transfer delay as a function of the transmitted file size when we change the access link bandwidth. We set C=100 Mbps, N=1500, D=20 msec (Figure 6) and 90 msec (Figure 7).

From both figures, we can observe that the packet loss ratio increases when the access link bandwidth increases. This is because the bursty nature increases and some of the bursty packet transmissions cannot be absorbed at the router buffer. However, the characteristics of the two disciplines would change drastically if we changed the propagation delay of the bottleneck link (D). When D is small (in Figure 6), the two disciplines have almost the same value for the packet loss ratio, and this causes an almost identical trend in the data transfer delay in Figure 6(b). When we increase D, however,

 ISBN # 1-56555-317-9

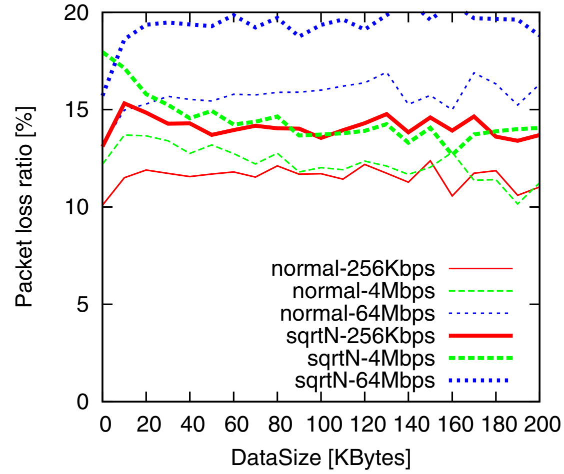


(a) Packet loss ratio

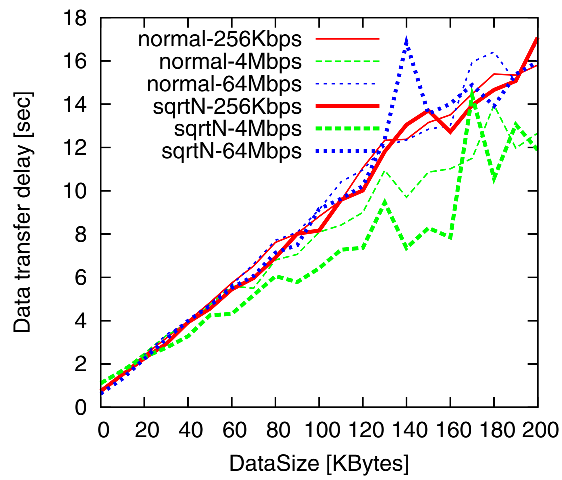


(b) Data transfer delay

Fig. 5. Effect of transferred data size (D=90 msec)



(a) Packet loss ratio



(b) Data transfer delay

Fig. 6. Effect of the bandwidth of the access link (D=20 msec)

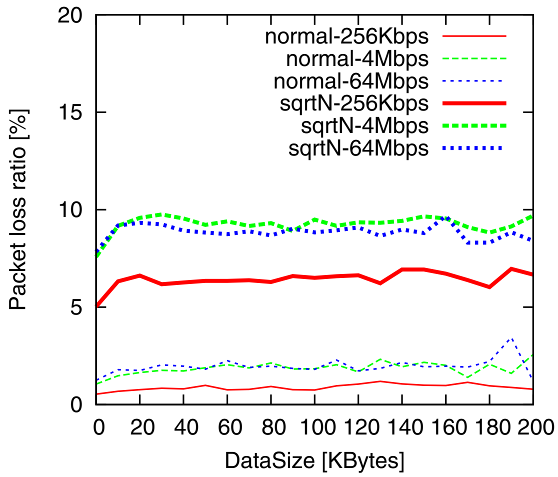
the sqrtN discipline has a quite larger value for the packet loss ratio compared with the normal discipline (Figure 7(a)) and the data transfer delay is affected by the difference in the packet loss ratio. This is because, when D is large, the buffer size in the normal discipline increases significantly, which can absorb the bursty packet arrivals from the TCP senders.

3) Bottleneck link propagation delay: We investigate last the effect of the propagation delay of the bottleneck link. Figure 8 shows the change of the packet loss ratio and the data transfer delay when the bottleneck link propagation delay is changed to $C = 100$ Mbps and $N = 1000$.

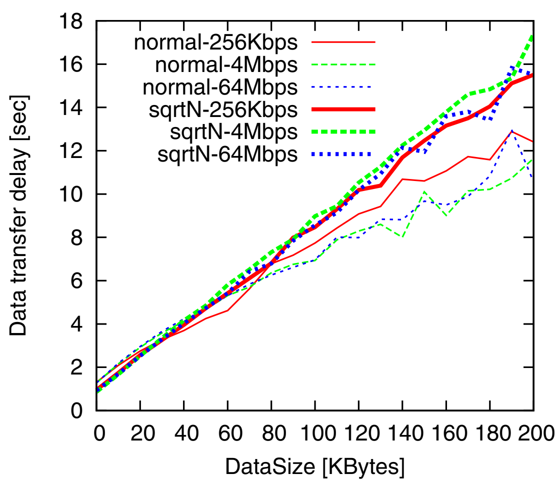
From this figure, we can also observe the higher packet loss ratio in the sqrtN discipline regardless of the propagation delay and transferred data size (Figure 8(a)). However, it does not always degrade the data transfer delay (Figure 8(b)). In particular, when either the propagation delay or the transferred data size is small, the data transfer delay remains almost the same as that in the normal discipline. When the propagation delay is small, the detection and retransmission of the lost packets in the network can be carried out in a small amount of time, which overcomes the increase in the packet loss ratio. When the transferred data size is small, on the other hand, the effect of the packet loss ratio becomes small, as described in the mathematical analysis in [4].

4) Bottleneck link bandwidth: Figure 9 shows the change in the packet loss ratio as a function of the bottleneck link bandwidth when we set D=20 msec (Figure 9(a)), and D=90 msec (Figure 9(b)). Here we set N=1500.

It can be seen that the link utilization with the sqrtN discipline is smaller than that with the normal discipline. Es-

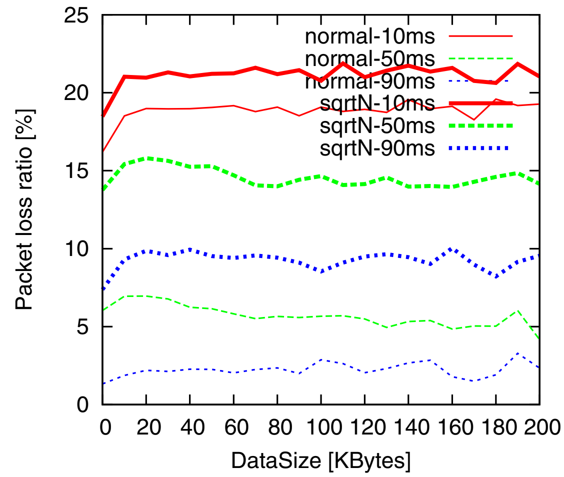


(a) Packet loss ratio

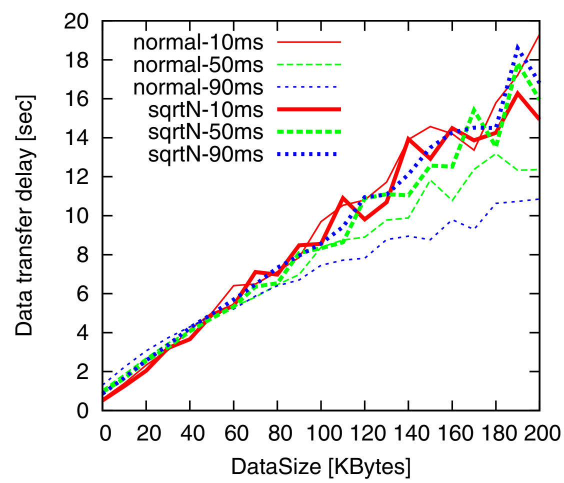


(b) Data transfer delay

Fig. 7. Effect of the bandwidth of the access link (D=90 msec)



(a) Packet loss ratio



(b) Data transfer delay

Fig. 8. Effect of the bottleneck link propagation delay

pecially, in Figure 9(a), the sqrtN discipline loses up to about 10% of the link bandwidth utilization when the bottleneck link bandwidth is large. These results mean that the sqrtN discipline would hinder the utilization of link bandwidth in the under-utilized network, whereas it can maintain the link utilization in the congested network. The main reason for this result is that the packet loss ratio in the sqrtN discipline never decreases to zero, even when the bottleneck link bandwidth is sufficiently large. This is one of the negative effects of a smaller-sized buffer at the bottleneck link.

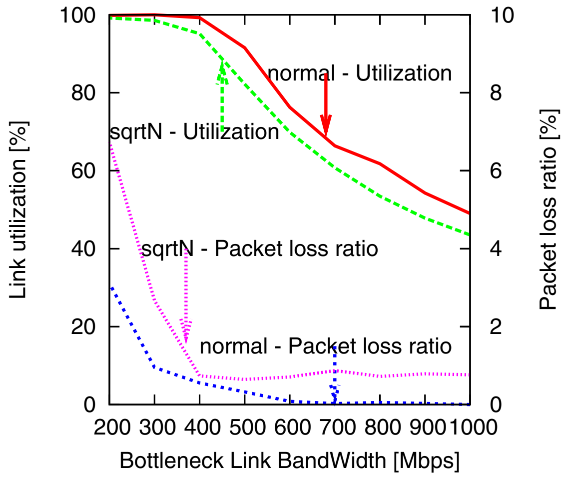
C. Summery

When buffer's size is determined by using a sqrtN discipline, if there are many flows, the utilization of the bottleneck link almost equal to the case of a normal discipline. but about the packet loss ratio of each flow, it becomes higher, and if the case where transferred data size nor a bottleneck link propagation delay are small, data transfer delay also becomes large.

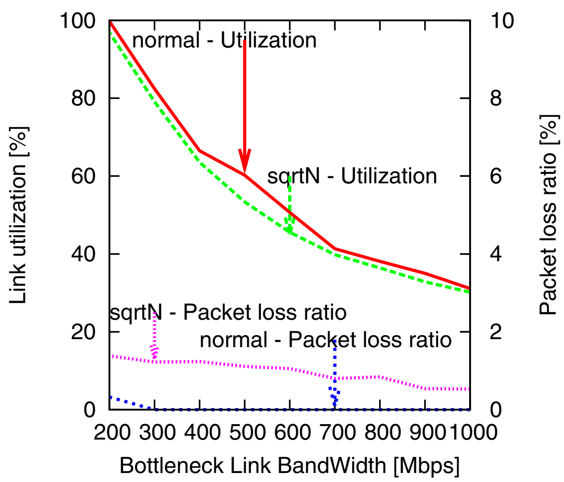
When network load is high, the influence of increasing the amount of traffic is not so large. but when network load is low, since the packet loss ratio does not fall, even if a bottleneck link bandwidth is large, compared with a normal discipline, the utilization of the link and a data transfer performance deteriorate.

From these things, sqrtN discipline is useful in a network which includes enough hosts in order to enlarge bottleneck load and small data occupies the greater part of the flow and/or the bottleneck link propagation delay is short.

However, on the present Internet, much traffic with large



(a) Effect of the bandwidth of the bottleneck link (D=20 msec)



(b) Effect of the bandwidth of the bottleneck link (D=90 msec)

Fig. 9. Packet loss ratio

data sizes, such as P2P, is also flowing. Moreover, as for flows other than TCP, such as UDP, the increase in a packet loss ratio has big influence on communicative quality, and a backbone network are designed so that an average utilization of the link may become low. From these things, It is conjectured to have a bad influence on the performance of TCP to use a sqrtN discipline under the present environment .

V. Conclusion

In this paper, we compared the performance of the above two disciplines for router buffer sizing, and we especially focused on the performance of TCP connections traversing the router. Through extensive simulations, we confirmed that the sqrtN discipline can maintain utilization of the bottleneck link when there is enough traffic volume for both long-lived and short-lived traffic flows. However, we revealed that the sqrtN discipline would degrade the performance of each TCP flow passing through the bottleneck link in terms of packet loss ratio and file transmission delay, and it can maintain each flow's performance only when the file transfer size is around 50-100 Kbytes or when the propagation delay between the sender and the receiver hosts is significantly small.

In the future, we will investigate the effect of pacing TCP on the buffer sizing problem. Furthermore, we will intensively study the conditions in which TCP connections sharing the bottleneck link behave synchronously, which would significantly affect the buffer sizing.

References

[1] J. Postel, "Transmission Control Protocol," *RFC 793*, Sep. 1981.
[2] J. Postel, "User Datagram Protocol," *RFC 768*, Oct. 1980.
[3] K. Mitsuya, K. Cho, A. Kato, and J. Murai, "IP traffic trends in packet size distribution," in *Proceedings of Internet conference 2000 (IC 2000)*, Nov. 2000.
[4] N. Cardwell, S. Savage, and T. Andreson, "Modeling TCP latency," in *Proceedings of IEEE INFOCOM 2000*, pp. 1742–1751, Mar. 2000.
[5] W. R. Stevens, *TCP/IP Illustrated, Vol 1: The Protocols*. Reading, Massachusetts: Addison-Wesley, 1994.
[6] C.Villamizar and C.Song, "High performance TCP in ANSNET," *SIGCOMM Computer Communications Review*, vol. 24, pp. 45–60, Oct. 1994.
[7] R. Bush and D. Meyer, "Some Internet architectural guidlines and philosophy," *RFC 3439*, Dec. 2003.
[8] G. Appenzeller, I. Keslassy, and N. McKeown, "Sizing router buffers," in *Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols For Computer Communications*, Sep. 2004.
[9] C. J. Fraleigh, *Provisioning Internet backbone Networks to support latency sensitive applications*. PhD thesis, Stanford University, Department of Electrical Engineering, June 2002. Adviser-Fouad A. Tobagi.
[10] M. Enachescu, Y. Ganjali, A. Goel, N. McKeown, and T. Roughgarden, "Part III: Routers with very small buffers," *SIGCOMM Computer Communications Review*, vol. 35, pp. 83–90, July 2005.
[11] S. S. A. Aggarwal and T. Anderson, "Understandinf the performance of TCP pacing," in *Proceeding of IEEE INFOCOM 2000 Conference on Computer Communications*, Mar. 2000.
[12] A. Dhamdhere and C. Dovrolis, "Open issues in router buffer sizing," *ACM SIGCOMM Computer Communication Review*, vol. 36, pp. 87–92, Jan. 2006.
[13] V. Jacobson, "Modified TCP congestion control argorithm," *End2end-interest mailing list*, Apr. 1990.
[14] S. Floyd and T. Henderson, "The NewReno modification to TCP's cfast recovery algorithm," *RFC 2582*, Apr. 1999.
[15] S. Shenker, L. Zhang, and D. Clark, "Some observation on the dynamics of a congestion control algorithm," *ACM Computer Communications Review*, vol. 20, pp. 30–39, Oct. 1990.
[16] Y. J. A. Gilbert and N. McKeown, "Congestion control and periodic behavior," in *Proceeding of LANMAN Workshop*, Mar. 2001.
[17] S. Floyd and V.Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397–413, 1993.
[18] L. Zhang and D. Clark, "Oscillating behavior of network traffic: A case study simulation," *Internetworking: Research and experience*, vol. 1, no. 2, pp. 101–112, 1990.
[19] L. Qiu, Y. Zhang, and S. Keshav, "Understanding the performance of many TCP flows," *Computer Networks*, vol. 37, no. 3-4, pp. 277–306, 2001.
[20] G. Iannaccone, M. May, and C. Diot, "Aggregate traffic performance with active queue management and drop from tail," *ACM Computer Communication Review*, vol. 31, pp. 4–13, July 2001.
[21] T. V. Project, "UCB/LBNL/VINT network simulator - ns (version 2)." available at `http://www.isi.edu/nsnam/ns/`.
[22] P. Barford and M. Crovella, "Generating representative Web workloads for network and server performance evaluation," in *Proceedings of the ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems*, pp. 151–160, July 1998.