Volume 31, issue 16, 25 October 2008 ISSN 0140-3664

ELSEVIER

computer communications

Available online at ScienceDirect
www.sciencedirect.com

Special Issue:
Performance Evaluation of Communication Networks (SPECTS 2007)
Guest Editor: Mohammad S. Obaidat

Computer Communications 31 (2008) 3789–3798

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Simulation studies on router buffer sizing for short-lived and pacing TCP flows

Go Hasegawa *, Takeshi Tomioka, Kentarou Tada, Masayuki Murata

Graduate School of Information Science and Technology, Osaka University, 1-32 Yamadaoka, Suita, Osaka 565-0871, Japan

ARTICLE INFO

Article history:
Available online 11 May 2008

Keywords:
Router
Buffer size
TCP
Bandwidth–delay product
Pacing TCP

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 primarily for long-lived flows. In contrast, there has not been a thorough verification of short-lived flows, which make up the majority of Internet flows. Furthermore, the effects of network parameters, such as the link bandwidth and propagation delay, have not yet been investigated. In the present paper, we compare the performance of the above two disciplines by simulation experiments. We focus on the performance of both long-lived and short-lived TCP connections traversing the router under various network environments. We show that 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 size of the file being transferred is approximately 50–100 Kbytes or when the propagation delay between the sender and the receiver hosts is significantly small. In addition, we demonstrate that using pacing TCP cannot improve the network performance in many situations and that sqrtN discipline is not suitable for situations in which pacing and non-pacing TCP flows co-exist in the network.

1. Introduction

At present, many applications rely on Transmission Control Protocol (TCP) to avoid and resolve congestion in the Internet. Although other applications utilize User Datagram Protocol (UDP) to control the network congestion on their own, on the current Internet, the proportion of UDP traffic is very small compared with TCP traffic [1]. Furthermore, in the current Internet, the traffic volume of video streaming applications such as YouTube increases [2] and they utilize TCP, not UDP, for a transport-layer protocol. 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 [3,4]. The output link buffer of almost all 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 be accumulated at this buffer, which causes queuing delay and delay jitter. Furthermore, packet losses also occur 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 because 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 [5]. As stated in [5], the size of a router's buffer should be greater than $B_n = C \times \text{RTT}$, that is, the product of the link bandwidth and the average RTT of flows that pass through the router. This is the bandwidth–delay product discipline (referred to herein as *normal discipline*), and many routers are equipped with buffers for which the size is determined by this discipline. This discipline is also described in a recent RFC [6].

However, according to [7], 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 have a link bandwidth of 2.5 Gbps or 10 Gbps [8]. If the average RTT equals 250 ms a 10 Gb/s router needs 250 ms × 10 Gbps = 2.5 Gbits for its buffer. The size of the largest commercial static RAM (SRAM) chip is currently 72 Mbits, which means that several dozen SRAM chips are needed to provide a 2.5-Gbps buffer. This results in large overhead in terms of board size, electrical power consumption, and monetary cost. On the other hand, the dynamic RAM (DRAM) chip is available up to 1 Gbps as well as significant advantages in monetary cost and board size. However, DRAM has a random access time of dozens of ns, which is from five times to ten 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 much larger than that of SRAM. In summary, it is

[*] Corresponding author. Tel.: +81 6 6850 6864; fax: +81 6 6850 6868.
E-mail address: hasegawa@ist.osaka-u.ac.jp (G. Hasegawa).

0140-3664/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2008.04.025

extremely difficult to build a router buffer for current and future high-speed networks based on normal discipline.

One possible solution for this problem is reported in [7]. 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, $B_s = \frac{C\times\overline{RTT}}{\sqrt{N}}$, when there are many flows (500 or more) passing through the link. We call this guideline *sqrtN discipline*. The authors in [7] assert that this small buffer size is sufficient to maintain the link utilization as well as that in normal discipline. In [9], the authors state that we need only dozens of packets for the router buffer size when the input link bandwidth is significantly smaller than the output link bandwidth (for example, 10 Mbps input and 1 Gbps output), and/or when using pacing TCP [10] (paced TCP), in which the successive data packets are transmitted with some time intervals to prevent the packets from being sent in bursts. Pacing TCP packets would decrease the packet loss ratio at the bottleneck router, which may contribute to the decrease in buffer size while maintaining the link utilization.

However, these studies consider only the utilization of the bottleneck link bandwidth as a performance metric 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 [9] cannot be satisfied in future networks: the link bandwidth of the access network is increasing rapidly in recent years.

Therefore, in the present 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. In particular, 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 the bottleneck link utilization, but also by factors including the RTT, packet loss ratio and available bandwidth. We investigate the effect of other network parameters such as the propagation delay and physical capacity of the bottleneck link, and derive the parameter ranges in which sqrtN discipline is effective or ineffective. In addition, we explore the effectiveness of pacing TCP for decreasing the router buffer, in situations in which only pacing TCP flows exists in the network and in which pacing and non-pacing TCP flows co-exist in the network.

To our knowledge, the effect of the router buffer size on the performance of short-lived TCP connections has only discussed in one paper [11], which revealed that the packet loss ratio becomes larger when we use the smaller-sized buffer recommended in [7], and it sometimes hinders the performance of TCP data transfer. However, the abovementioned study [11] was performed with a fixed network environment, and the authors only considered congested networks with approximately 100% link bandwidth utilization. On the other hand, in the present study, 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 that TCP connections transmit, unlike the fixed value for transferred file sizes used in [11].

We believe that for the complete comparison of the two discipline in buffer sizing, we should evaluate them from various points of view. It includes the effect of network parameters: the effect of RTT (propagation delay), access/bottleneck link bandwidth, with homogeneous/heterogeneous situation. It also includes the effect of various type of TCP flows: short/long-lived and paced/non-paced, and their mixture situation. Among them, we select the following cases in this paper: the effect of long/short-lived flow, the effect of paced TCP and its mixture situation, and the effect of network parameters with homogeneous situation. This is because we would like to reveal the fundamental characteristics of the two discipline.

The remainder of the paper is organized as follows. Section 2 reviews the two disciplines for determining router buffer size: normal discipline and 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 discuss the effect of pacing TCP in router buffer sizing. Section 6 concludes the present paper and gives some future areas of study.

2. Guidelines for router buffer sizing

2.1. Normal (bandwidth–delay product) discipline

The traditional guideline for setting the buffer size based on the bandwidth–delay product is described in [5]. We call this guideline normal discipline. In what follows, we introduce the fundamental reasons for normal discipline. For a detailed explanation, please refer to [5].

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 [12] and NewReno [13]. Fig. 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 to be 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). Approximately one RTT later, the sender receives duplicate ACKs. The sender then retransmits the lost packet, and halves its window size from $W_{\max}$ to $W_{\max}/2$ (at time t_3). Before time t_3, the sender is allowed to have $W_{\max}$ outstanding packets. However, after time t_3, the sender is only allowed to have $W_{\max}/2$ outstanding packets. Therefore, the sender must stop sending packets until it receives $W_{\max}/2$ ACK packets. This means that 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 its window size, so the number of packets in the buffer again increases after time t_5.

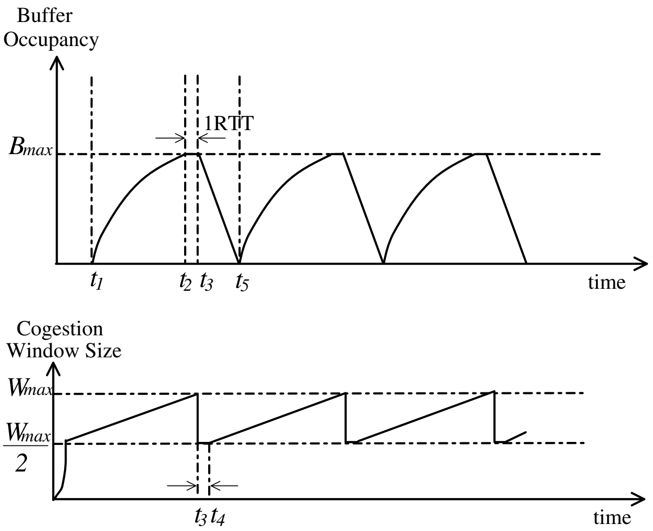

Fig. 1. The time evolution of the congestion window and the queue length.

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. Therefore, the sender stops sending packets for $(W_{\max}/2)/C$ s in order to wait for $W_{\max}/2$ ACK packets. On the other hand, the buffer is emptied after $B_{\max}/C$ s. 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 prevent the buffer from becoming empty:

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

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

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

Finally, Eqs. (1) and (2) yield

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

In summary, if $B_{\max} \geqslant C \times \overline{\mathrm{RTT}}$ 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 bottleneck link, we can consider that an individual flow has a bandwidth of C/N [14–17]. This means that each flow needs more than $C/N \times \overline{\mathrm{RTT}}$ for the buffer capacity. Therefore, the required buffer capacity can still be shown as follows:

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

2.2. Square-root discipline

In [7], the authors proposed 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 the present paper, we call this guideline *sqrtN discipline*.

The window sizes of TCP connections usually change synchronously when the number of concurrent connections is small and their RTTs are approximately equal [14–17]. This means that when a certain connection halves its window size, the others do the same simultaneously [14]. 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, in many cases, flows are not synchronized. For example, small variations in RTTs or processing times are sufficient to prevent synchronization [18]. The absence of synchronization has been demonstrated in real networks [8,19]. Even if flows do not have a diversity of RTTs, they can become asynchronous when there are more than 500 concurrent flows [7]. In what follows, we briefly introduce the required buffer size when TCP flows are not synchronized, which is the summary of the discussion in [7].

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

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

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:

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

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

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

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 a 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

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

From Eqs. (7)–(9) the standard deviation of the queue occupancy, Q, is shown as:

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

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

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

For example, if there are 10,000 concurrent flows, then $Util \geqslant erf\left(\frac{3\sqrt{3}}{2\sqrt{2}}\right) \simeq 0.9899$ when we set $B_{\max} = \frac{\overline{\mathrm{RTT}} \times C}{\sqrt{N}}$. This result means that we can achieve 98.99% of the link utilization with a buffer having a size that is given by the bandwidth–delay product divided by the square-root of the number of flows, that is, $B_s = \frac{\overline{\mathrm{RTT}} \times C}{\sqrt{N}}$. In [7], the effectiveness of sqrtN discipline is confirmed by simulations and experiments, but consideration 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 [7]. 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 small buffer size on short-lived flows through extensive simulations.

3. Evaluation environment

3.1. Network and traffic model

We evaluate the performance of the two disciplines for buffer sizing using ns-2 [20] simulations. The network model used for the simulations is shown in Fig. 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 ms propagation delay and C Mbps bandwidth. The links between the terminals and routers have a 5 ms propagation delay and bandwidth equal to the bottleneck link if not specified. We vary N, C, D, and the access link bandwidth in the simulations and investigate the performance of the two buffer sizing disciplines.

We use two types 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 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

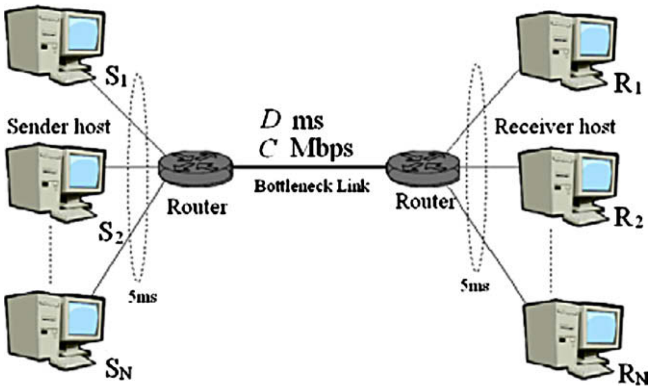

Fig. 2. Network topology for simulation experiments.

intervals based on the Scalable URL Reference Generator (SURGE) model [21]. SURGE is a realistic Web workload generation tool that mimics a set of real users accessing a server.

Table 1 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).

3.2. 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.

4. Simulation results and discussions

4.1. Basic performance

Fig. 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 (Fig. 3(a)) and the Web traffic model (Fig. 3(b)). We set $C = 100$ Mbps and $D = 90$ ms.

Fig. 3(a) shows that when the buffer size is determined by 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, when 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 normal discipline. However, we conclude that the assertion in [7] is correct because the link utilization is almost 100% when there is a sufficiently large 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.

Table 1
Summary statistics for models used in SURGE [21].

Component	Probability density function	Parameters
File sizes – body	$p(x) = \frac{e^{-(\ln x-\mu)^2/2\sigma^2}}{x\sigma\sqrt{2\pi}}$	$\mu = 9.357$ $\sigma = 1.318$
File sizes – tail	$p(x) = \alpha k^{\alpha} x^{-\alpha+1}$	$k = 133\ K$ $\alpha = 1.1$
Inactive OFF times	$p(x) = \alpha k^{\alpha} x^{-\alpha+1}$	$k = 1$ $\alpha = 1.5$

In Fig. 3(b) for Web traffic, we can observe that the link utilization with sqrtN discipline is also lower than that with normal discipline when short-lived TCP flows are accommodated. Note that the link utilization with sqrtN discipline becomes degraded in under-utilized networks; even when the link utilization with normal discipline is around 60–80%, 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 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 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 [7] is also confirmed even with short-lived Web traffic.

In the following, we investigate whether the conclusion in [7] holds even when the network environment changes, and we check the characteristics of sqrtN discipline in terms of the performance of each TCP flow passing through the router. Note that we omit to plot the confidence intervals, but it is enough small to make a comparison by using average values.

4.2. Effect of the change in network environment

We next discuss whether normal discipline or 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.

4.2.1. Traffic volume

Figs. 4 and 5 show 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 use the Web traffic model for each sender host, and set $C = 100$ Mbps and $D = 20$ ms (Fig. 4) and 90 ms (Fig. 5).

Figs. 4(a) and 5(a) show that the packet loss ratio with sqrtN discipline is always higher than that with normal discipline. This is confirmed by Fig. 3 in the previous subsection. We also point out that the packet loss ratio with sqrtN discipline increases as the transferred file size decreases when $D = 20$ ms, whereas that with 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 sqrtN discipline cannot absorb the burstiness.

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

4.2.2. Access link bandwidth

Figs. 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

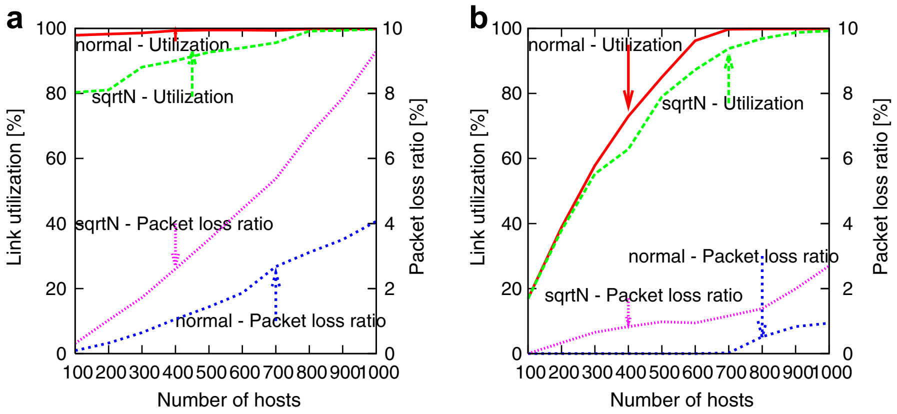

Fig. 3. Effect of buffer size on link utilization and packet loss ratio. (a) P2P traffic. (b) Web traffic.

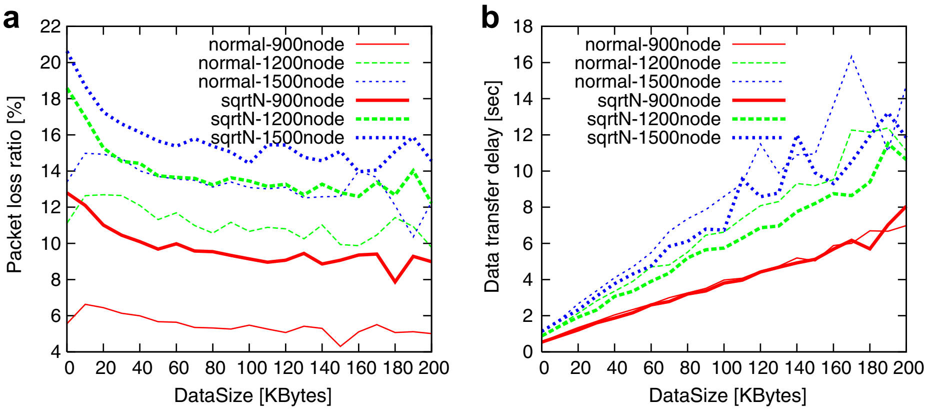

Fig. 4. Effect of transferred data size ($D = 20$ ms). (a) Packet loss ratio. (b) Data transfer delay.

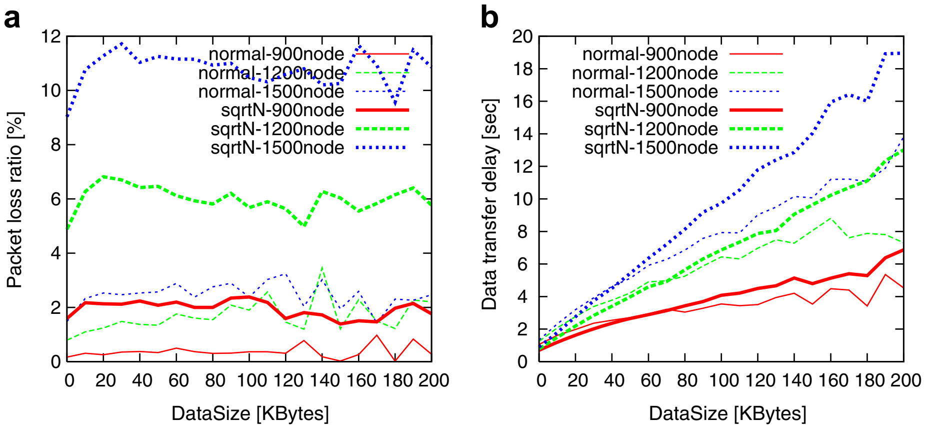

Fig. 5. Effect of transferred data size ($D = 90$ ms). (a) Packet loss ratio. (b) Data transfer delay.

change the access link bandwidth. We set $C = 100$ Mbps, $N = 1500, D = 20$ ms (Fig. 6) and 90 ms (Fig. 7).

Both figures show that the packet loss ratio increases as 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 Fig. 6), the two disciplines have almost the same packet loss ratio, and this causes an almost identical trend in the data transfer delay in Fig. 6(b), which can be recognized by comparing the two lines of sqrtN and normal disciplines. When we increase D, however, sqrtN discipline has a much larger packet loss ratio compared with normal discipline (Fig. 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 normal discipline increases significantly, which can absorb the bursty packet arrivals from the TCP senders.

4.2.3. Bottleneck link bandwidth

Fig. 8 shows the change in the packet loss ratio as a function of the bottleneck link bandwidth when we set $D = 20$ ms (Fig. 8(a)), and $D = 90$ ms (Fig. 8(b)). Here we set $N = 1500$.

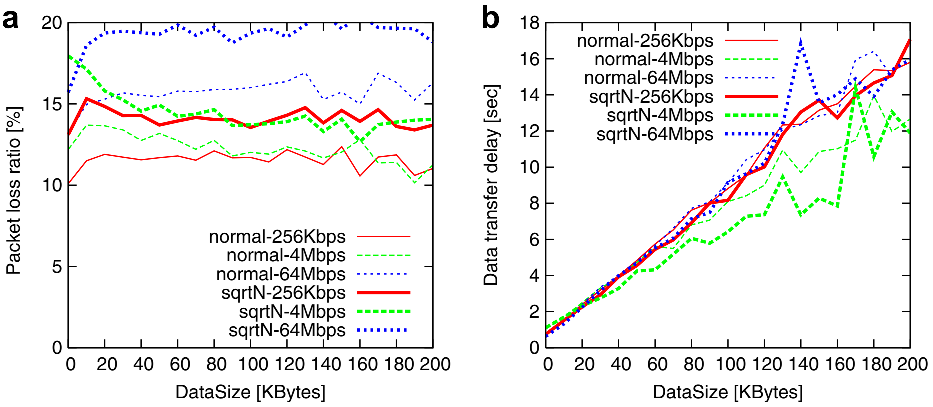

Fig. 6. Effect of the bandwidth of the access link ($D = 20$ ms). (a) Packet loss ratio. (b) Data transfer delay.

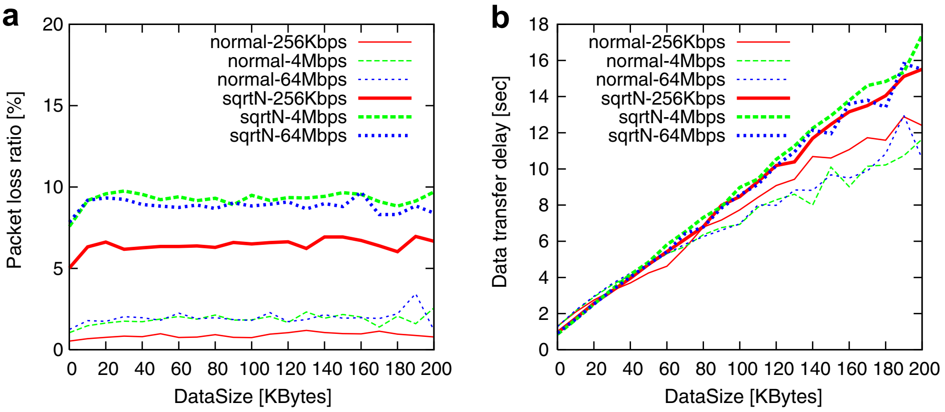

Fig. 7. Effect of the bandwidth of the access link ($D = 90$ ms). (a) Packet loss ratio. (b) Data transfer delay.

The link utilization with sqrtN discipline is smaller than that with normal discipline. In particular, in Fig. 8(a), sqrtN discipline loses up to approximately 10% of the link bandwidth utilization when the bottleneck link bandwidth is large. These results mean that sqrtN discipline would hinder the utilization of the link bandwidth in an under-utilized network, whereas it can maintain the link utilization in a congested network. The main reason for this result is that the packet loss ratio in sqrtN discipline never decreases to zero, even when the bottleneck link bandwidth is sufficiently large. This is one of the adverse effects of a smaller buffer at the bottleneck link.

4.2.4. Bottleneck link propagation delay

Finally, we investigate the effect of the propagation delay of the bottleneck link. Fig. 9 shows the change in 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 sqrtN discipline regardless of the propagation delay and transferred data size (Fig. 9(a)), which is obvious by comparing the two lines of sqrtN and normal disciplines. However, this does not always degrade the data transfer delay (Fig. 9(b)). In particular, when either the propagation delay or the transferred data size is

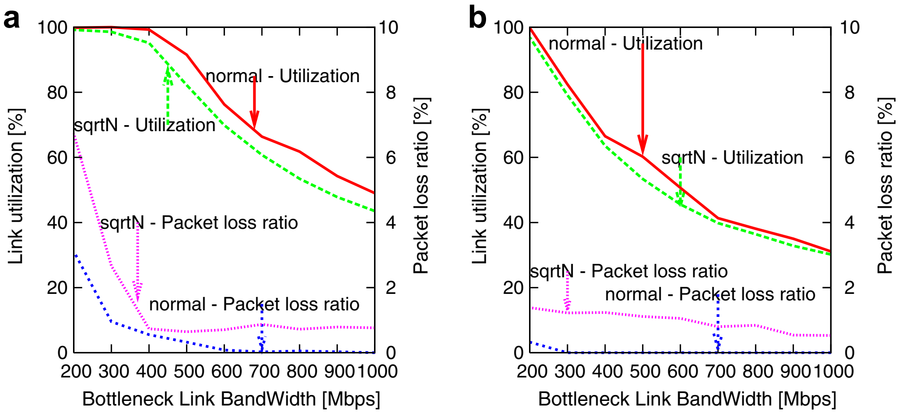

Fig. 8. Packet loss ratio. (a) Effect of the bottleneck link bandwidth ($D = 20$ ms). (b) Effect of the bottleneck link bandwidth ($D = 90$ ms).

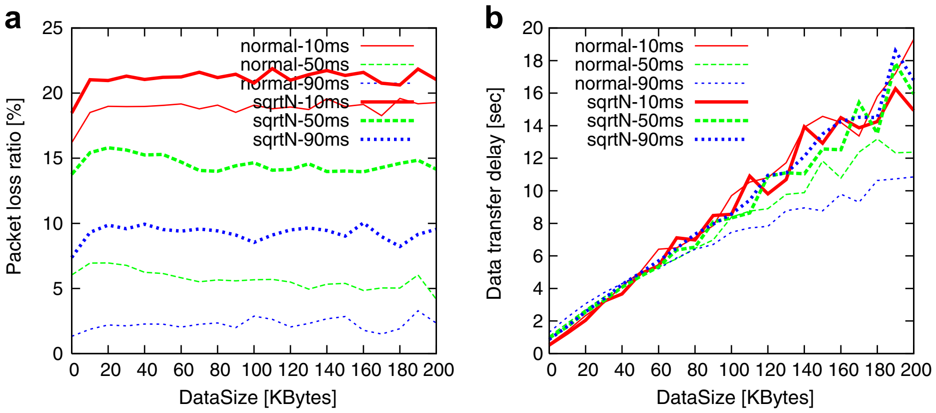

Fig. 9. Effect of the bottleneck link propagation delay. (a) Packet loss ratio. (b) Data transfer delay.

small, the data transfer delay remains almost the same as that in 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 [3].

4.3. Summary

When the buffer size is determined using a sqrtN discipline, if there are many flows, the utilization of the bottleneck link is approximately equal to the case of a normal discipline. However, the packet loss ratio of each flow becomes higher, and only for the case in which either the transferred data size or a bottleneck link propagation delay are small, the data transfer delay also becomes small.

When the network load is high, the influence of increasing the amount of traffic is not so significant. However, when the 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. This causes the deterioration of the link utilization and data transfer performance. Therefore, sqrtN discipline is useful in a network that includes a sufficient number of hosts so that the bottleneck link load is enlarged and small-size data transmission occupies the greater part of the flow and/or the bottleneck link propagation delay is short.

However, on the present Internet, significant amount of traffic with large data sizes, such as P2P, is also exists. Moreover, for flows other than TCP, such as UDP, the increase in a packet loss ratio has a significant influence on the communication quality, and a core network of today's Internet is designed so that an average link utilization keep low. Therefore, it is assumed that the use of sqrtN discipline under the present Internet environment will have an adverse effect on the performance of TCP.

5. Effect of pacing TCP

We next investigate the effect of pacing TCP on router buffer sizing. For simulation experiments, we utilize the same network model (Fig. 2) with $C = 100$ Mbps and $D = 90$ ms. For TCP flows we deploy the P2P traffic model explained in Section 3.

5.1. Link utilization and packet loss ratio

Fig. 10 shows the change in bottleneck link utilization and packet loss ratio as functions of n (the number of concurrent TCP flows), when we utilize pacing TCP with two disciplines for buffer sizing. For comparison, we also plot the results when we utilize non-pacing TCP. From Fig. 10(a), when we use non-pacing TCP flows with sqrtN discipline, the link utilization increases to 100% with the increase of n, as depicted in Fig. 3(a). However, when we use pacing TCP, the link utilization never reach to 100% even when the number of concurrent TCP flows increases significantly. From Fig. 10(b), we also observe that the effect of pacing TCP in decreasing the packet loss ratio at the bottleneck router is quite limited, especially with sqrtN discipline.

Fig. 11 explains the reasons for this phenomena. In this figure, we plot, using dots, the occurrence of packet loss events for each TCP connection as a function of simulation time when we set

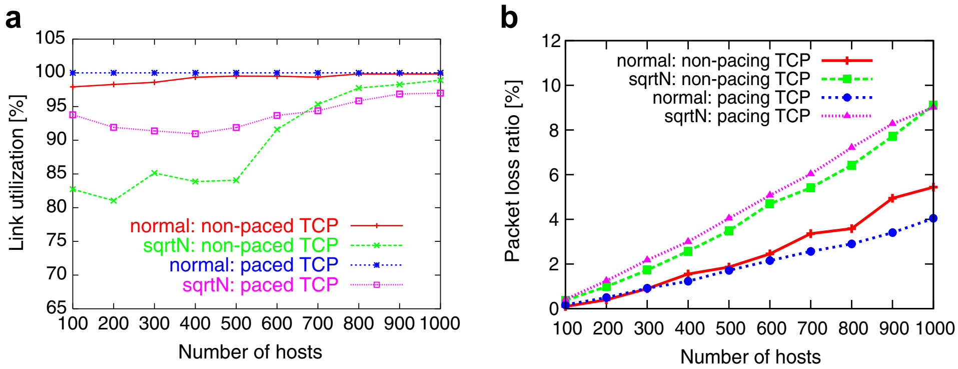

Fig. 10. Effect of pacing TCP. (a) Link utilization. (b) Packet loss ratio.

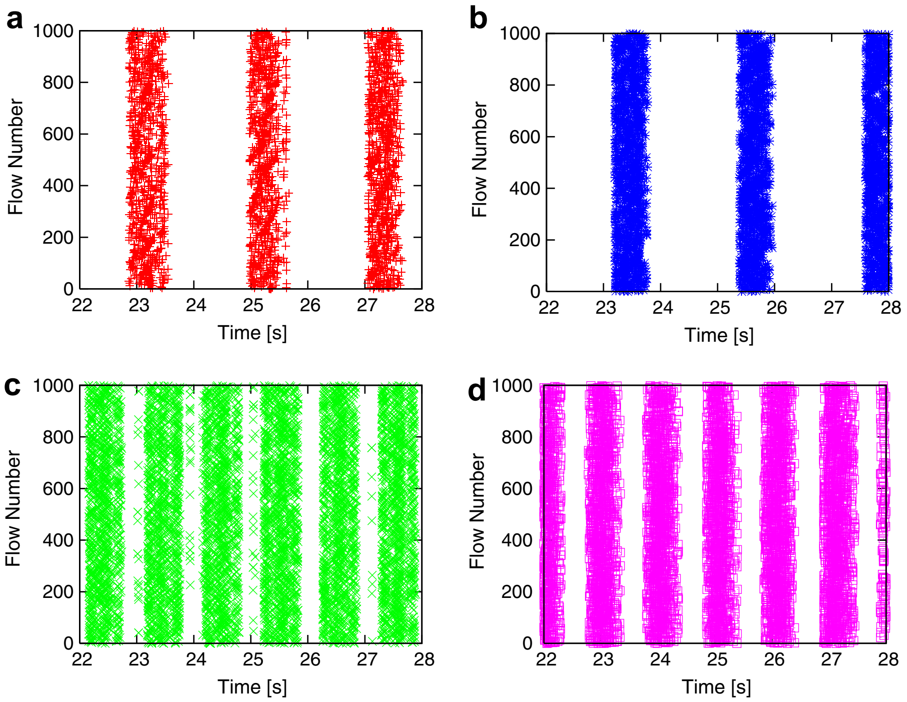

Fig. 11. Packet loss events. (a) Non-pacing TCP with normal discipline. (b) Pacing TCP with normal discipline. (c) Non-pacing TCP with sqrtN discipline. (d) Pacing TCP with sqrtN discipline.

$n = 1000$. In all four cases of the combination of TCP type (pacing/non-pacing) and buffer size disciplines (normal/sqrtN), packet losses takes place in synchronized fashion. Furthermore, by comparing Figs. 11(a) and (b) and 11(c) and (d), we can confirm that using pacing TCP increases the degree of synchronization of packet loss events, which degrades the link utilization. We can explain this phenomena as follows:

When TCP flows are non-paced, the packers from each flow tend to arrive at the bottleneck link in bursty fashion. So, the queue length fluctuate largely and buffer overflows often occurs even when the buffer is not fully utilized in average. So, "unfortunate" TCP flow(s) experiences the packet loss(es) in the earlier stage of the congestion, and the synchronization effect does not become so strong.

On the other hand, when the TCP flows are paced, the packers from each flow are sent in non-bursty fashion, and the packers from all flows arrives at the bottleneck link in mixed fashion. So, the queue length does not fluctuate so largely and it increases in steady speed. Then, when buffer overflow occurs, almost all of the TCP flows experience packet losses simultaneously as in Fig. 11. It causes the global synchronization effect and degrades the link utilization as shown in Fig. 10.

5.2. Mixture of pacing and non-pacing TCP flows

We next observe the situation in which pacing and non-pacing TCP flows co-exist in the network. Fig. 12 shows the change in bottleneck link utilization and packet loss ratio as functions of the number of pacing TCP flows when we set the total number of TCP flows (n) to 100 (Fig. 12(a)) and 1000 (Fig. 12(b)). Note that we maintain the total number of TCP flows constant and change the ratio of the number of pacing and non-pacing TCP flows.

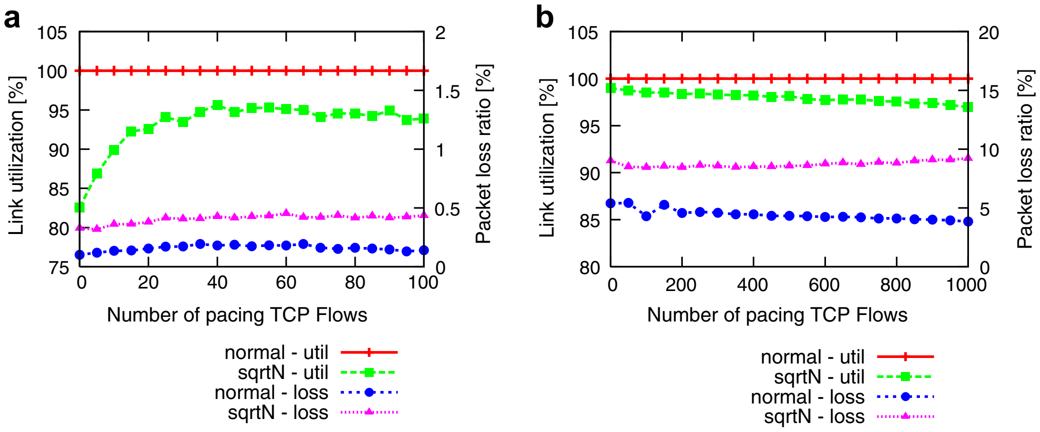

Fig. 12. Evaluation of mixture situation. (a) $n = 100$. (b) $n = 1000$.

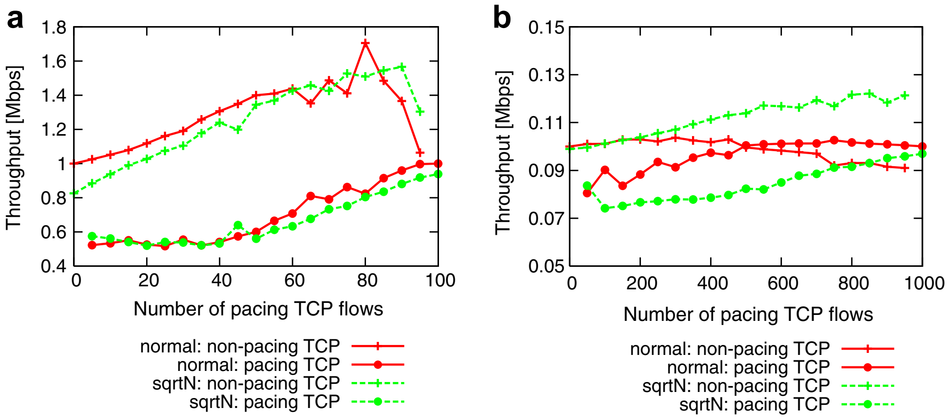

Fig. 13. Evaluation of mixture situation. (a) $n = 100$. (b) $n = 1000$.

Fig. 12 reveals that when using normal discipline for buffer sizing, the link utilization remains 100% in all situations. However, when we use sqrtN discipline, the link utilization depends on the total number of TCP flows and the ratio of pacing TCP flows. More specifically, when the total number of flows is small (Fig. 12(a)), an increase in that number of pacing TCP flows does not cause an increase in the packet loss ratio, which improves the link utilization. However, when the network is congested with many concurrent flows (Fig. 12(b)), the link utilization decreases with the increase in the ratio of pacing TCP flows because of the increase in the packet loss ratio. That is, using pacing TCP in the mixed flow situation does not help to increase the link utilization or decrease the packet loss ratio in the network.

Fig. 13 shows the changes in the throughput of each TCP flow when we set the total number of TCP flows (N) to 100 (Fig. 13(a)) and 1000 (Fig. 13(b)), respectively. When the total number of TCP flows is small (Fig. 13(a)), the throughput of non-pacing TCP flows is roughly twice that of pacing TCP flows for almost all cases, except when the ratio of pacing TCP flows is larger than 90%. Focusing on sqrtN discipline, when all of the flows use pacing TCP, the overall performance is larger than that for the case in which all flows use non-pacing TCP. However, in the mixture situation, pacing TCP flows suffer from significantly smaller throughput than non-pacing TCP flows.

On the other hand, when the number of total TCP flows increases to 1000 and the network becomes congested (Fig. 13(b)), the throughput of pacing TCP flows becomes larger than that of non-pacing TCP flows when the ratio of pacing TCP flows is larger than 50% and the buffer size is determined by normal discipline. However, when we deploy sqrtN discipline, pacing TCP flows never outperforms non-pacing TCP flows regardless of the ratio of pacing TCP flows.

Based on these results, we conclude that it is difficult to deploy pacing TCP to the current Internet when considering the mixture situation, although the throughput of each TCP flow improves when all flows utilize pacing TCP. Furthermore, using sqrtN discipline makes the situation worse for the deployment of pacing TCP.

6. Conclusion

In the present paper, we compared the performance of two disciplines, normal discipline and sqrtN discipline, for router buffer sizing, focusing on the performance of TCP connections traversing the router. Through extensive simulations, we confirmed that sqrtN discipline can maintain utilization of the bottleneck link when there is sufficient traffic volume for both long-lived and short-lived traffic flows. However, we revealed that 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. Furthermore, sqrtN discipline can maintain the performance of each flow 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.

We also found that using pacing TCP increases the degree of synchronization of packet loss events, which degrades the network performance in terms of bottleneck link utilization with sqrtN discipline. Furthermore, in the case of mixed situation of pacing and non-pacing TCP flows, the pacing TCP flows suffer from significantly lower throughput than the co-existing non-pacing TCP flows especially when sqrtN discipline is employed.

For future work, the evaluation in more heterogeneous situation, where short/long-lived flows co-exists in the network, where each TCP flows has different network parameters, and whrer TCP and UDP flows co-exist in the network, is one of our important issue. We will also study the conditions in which TCP connections sharing a bottleneck link behave synchronously, which could significantly affect the buffer sizing.

References

[1] K. Mitsuya, K. Cho, A. Kato, J. Murai, IP traffic trends in packet size distribution, in: Proceedings of Internet Conference 2000 (IC 2000), November 2000.

[2] The YouTube effect: HTTP traffic now eclipses P2P, Available from: <http://arstechnica.com/news.ars/post/20070619-the-youtube-effect-http-traffic-now-eclipses-p2p.html/>.

[3] N. Cardwell, S. Savage, T. Andreson, Modeling TCP latency, in: Proceedings of IEEE INFOCOM 2000, March 2000, pp. 1742–1751.

[4] W.R. Stevens, TCP/IP Ilustrated, Vol. 1: The Protocols. Reading, Addison-Wesley, MA, 1994.

[5] C. Villamizar, C. Song, High performance TCP in ANSNET, SIGCOMM Computer Communications Review 24 (1994) 45–60.

[6] R. Bush, D. Meyer, Some internet architectural guidelines and philosophy, RFC 3439, December 2003.

[7] G. Appenzeller, I. Keslassy, N. McKeown, Sizing router buffers, in: Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, September 2004.

[8] 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).

[9] M. Enachescu, Y. Ganjali, A. Goel, N. McKeown, T. Roughgarden, Part iii: Routers with very small buffers, SIGCOMM Computer Communications Review 35 (2005) 83–90.

[10] S.S.A. Aggarwal, T. Anderson, Understanding the performance of TCP pacing, in: Proceeding of IEEE INFOCOM 2000 Conference on Computer Communications, March 2000.

[11] A. Dhamdhere, C. Dovrolis, Open issues in router buffer sizing, ACM SIGCOMM Computer Communication Review 36 (2006) 87–92.

[12] V. Jacobson, Modified TCP congestion control algorithm, End2end-interest Mailing List, April 1990.

[13] S. Floyd, T. Henderson, The NewReno modification to TCP's cfast recovery algorithm, RFC 2582, April 1999.

[14] S. Shenker, L. Zhang, D. Clark, Some observation on the dynamics of a congestion control algorithm, ACM Computer Communications Review 20 (1990) 30–39.

[15] Y.J.A. Gilbert, N. McKeown, Congestion control and periodic behavior, in: Proceeding of LANMAN Workshop, March 2001.

[16] S. Floyd, V. Jacobson, Random early detection gateways for congestion avoidance, IEEE/ACM Transactions on Networking 1 (4) (1993) 397–413.

[17] L. Zhang, D. Clark, Oscillating behavior of network traffic: a case study simulation, Internetworking: Research and Experience 1 (2) (1990) 101–112.

[18] L. Qiu, Y. Zhang, S. Keshav, Understanding the performance of many TCP flows, Computer Networks 37 (3–4) (2001) 277–306.

[19] G. Iannaccone, M. May, C. Diot, Aggregate traffic performance with active queue management and drop from tail, ACM Computer Communication Review 31 (2001) 4–13.

[20] T.V. Project, UCB/LBNL/VINT network simulator – ns (version 2). Available from: <http://www.isi.edu/nsnam/ns/>.

[21] P. Barford, 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.