Understanding TCP over TCP: Effects of TCP Tunneling on End-to-End Throughput and Latency

Osamu Honda[a] and Hiroyuki Ohsaki[a] and Makoto Imase[a] and Mika Ishizuka[b] and Junichi Murayama[b]

[a] Graduate School of Infomation Science and Technology, Osaka University, Suita Osaka, Japan;
[b]NTT Information Sharing Platform Laboratories, NTT Corporation, Musashino Tokyo, Japan

ABSTRACT

TCP tunnel is a technology that aggregates and transfers packets sent between end hosts as a single TCP connection. By using a TCP tunnel, the fairness among aggregated flows can be improved and several protocols can be transparently transmitted through a firewall. Currently, many applications such as SSH, VTun, and HTun use a TCP tunnel. However, since most applications running on end hosts generally use TCP, two TCP congestion controls (i.e., end-to-end TCP and tunnel TCP) operate simultaneously and interfere each other. Under certain conditions, it has been known that using a TCP tunnel severely degrades the end-to-end TCP performance. Namely, it has known that using a TCP tunnel drastically degrades the end-to-end TCP throughput for some time, which is called *TCP meltdown* problem. On the contrary, under other conditions, it has been known that using a TCP tunnel significantly improves the end-to-end TCP performance. However, it is still an open issue — how, when, and why is a TCP tunnel malicious for end-to-end TCP performance? In this paper, we therefore investigate effect of TCP tunnel on end-to-end TCP performance using simulation experiments. Specifically, we quantitatively reveal effects of several factors (e.g., the propagation delay, usage of SACK option, TCP socket buffer size, and sender buffer size of TCP tunnel) on performance of end-to-end TCP and tunnel TCP.

Keywords: TCP (Transmission Control Protocol), TCP over TCP, TCP Tunnel, Performance Evaluation, Goodput, Round-Trip Time

1. INTRODUCTION

TCP tunnel is technology of building a virtual circuit, by aggregating flows (e.g., sequence of packets with the same source/destination IP address and port number) between two nodes (e.g. end host or router) and transfering them as a single TCP connection. With a TCP tunnel, fairness between aggregated flows can be improved and a firewall can be utilized transparently.[1, 2] TCP tunnel has been widely used in several tunneling applications, such as Vtun,[3] Htun,[4] SSH,[5] and SoftEther.[6]

In the literatures, a great number of studies regarding performance evaluation of TCP have been performed. [7-11] Several researches regarding performance evaluation of end-to-end TCP with a TCP tunnel have also been done. [1, 9, 12] For instance, the authors of the literatures[1, 9, 12] point out that the performance of end-to-end TCP degrades under certain conditions. On the contrary, they also point out that the performance of end-to-end TCP improves under other conditions. [9] However, in these researches, effect of several factors, such as network parameters and TCP parameter configurations, on the end-to-end TCP performance has not been sufficiently investigated.

In this paper, we investigate the effect of a TCP tunnel on the end-to-end TCP performance. In particular, we investigate the effect of a TCP tunnel on the goodput and the round-trip time of end-to-end TCP flows (for bevity, end-to-end TCP flow is simply called *TCP flow*). Furthermore, we investigate how TCP parameters should be configured when using a TCP tunnel to avoid degradation of the end-to-end TCP performance, given several factors such as network parameters.

Factors that may affect the end-to-end TCP performance include: network parameters such as the link bandwidth, the propagation delay, MTU (Maximum Transmission Unit) and the router buffer size; network workload such as the number

E-mail:[a] {o-honda,oosaki,imase}@ics.es.osaka-u.ac.jp, [b] {ishizuka.mika, murayama.junichi}@lab.ntt.co.jp

of TCP flows, the number of TCP tunnels, the traffic pattern of TCP flows and the background traffic; TCP configurations such as TCP version (e.g., Tahoe, Reno, New Reno and Vegas), existence of the SACK option, the socket buffer size and the initial value of RTO (Retransmission Time Out).

In this paper, we evaluate the performance of a TCP tunnel, focusing on factors that past studies on a TCP tunnel have not take account of. Specifically, as network parameters, we focus on the propagation delay of a link, and the buffer size of the ingress router of a TCP tunnel. Moreover, as TCP configuration parameters, we focus on existence of the SACK option and the socket buffer size

The organization of this paper is as follows. First, Section 2 explains related work on a TCP tunnel. In Section 3, after describing the simulation configuration, we show simulation results for quantitatively investigating effect of a TCP tunnel on the end-to-end TCP performance. Finally, Section 4 summarizes this paper and discusses future works.

2. RELATED WORKS

In the literature,[1] the authors have proposed a mechanism that realizes fairness between TCP and UDP traffic by accommodating them in different TCP tunnels when TCP and UDP share the bottleneck link. The authors have shown that using a TCP tunnel increased the round-trip time of a TCP flow approximately 4 times, and the goodput of a TCP flow was decreased by approximately 60%. It is known that the goodput of a TCP flow is degraded when there is a network with a small MTU on the path. However, the authors of the literature[1] have shown that using a TCP tunnel limited such performance degradation to approximately 34%.

However, in the literature,[1] effect of the link bandwidth, the propagation delay of a link, the buffer size of the ingress router of a TCP tunnel have not been investigated. Moreover, effect of the number of TCP flows, the number of TCP tunnels, the traffic pattern of TCP flows, the version of TCP, existence of the SACK option, the socket buffer size, and the initial value of RTO have not also been investigated.

In the literature,[9] the authors have investigated the effect of a TCP tunnel on the end-to-end TCP performance when a satellite circuit existed on the path of TCP flows. The authors of the literature[9] have shown that using a TCP tunnel decreased the goodput of TCP flows by approximately 10%. Their explanation is that the retransmission control of TCP causes such decrease in goodput of TCP flows. Moreover, it have been shown that using a TCP tunnel with the SACK option increased the goodput of TCP flows by approximately 45%. Furthermore, it has been also shown that the goodput of TCP flows improved up to 30% when the socket buffer size of a TCP tunnel was roughly equal to the bandwidth-delay product of the network,

However, in the literature,[9] effect of the link bandwidth, the propagation delay of a link, and the buffer size of the ingress router of a TCP tunnel have not been investigated. Also, effect of the number of TCP flows, the number of TCP tunnels, the traffic pattern of TCP flows, the version of TCP, existence of the SACK option, the socket buffer size, and the initial value of RTO have not been investigated.

The authors of the literature[12] explain that the RTO value used by the retransmission control of TCP causes degradation of the goodput of TCP flows. Namely, when the RTO of the end-to-end TCP is smaller than that of the tunnel TCP, the end-to-end TCP retransmits lost packets at a higher rate than the speed that the tunnel TCP can process. They suggest that such mismatch in RTO values of end-to-end and tunnel TCPs causes the decrease in the goodput of TCP flows. However, it is merely qualitative explanation and it has not been clarified in what conditions such performance degradation occurs.

3. SIMULATION

3.1. Simulation configuration

The network topology used in simulations is shown in Fig. 1. Simulations were performed while changing the propagation delay of the access link and backbone link in Fig. 1. TCP tunnel was established between the ingress and egress routers, and TCP traffic was continuously transmitted from the source host to the destination host. We implemented a TCP tunnel module in OPNET modeler 10.0A.[13]

The parameter configuration used in simulation is dsummaried in Tab. 1. In the following simulations, unless explicitly stated, parameters shown in Tab. 1 were used. As the background traffic, UDP traffic was generated on the backbone link. The packet arrival rate of the background traffic was given by the exponential distribution with the average arrival rate of

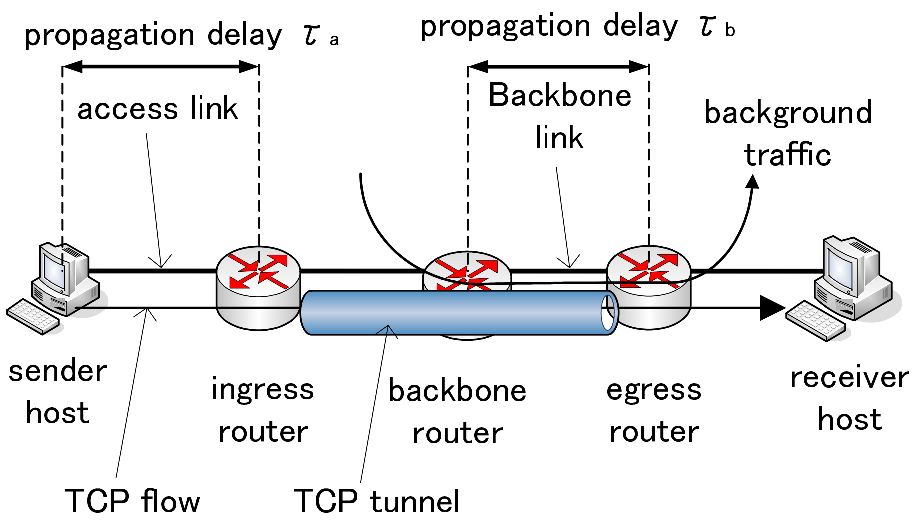

Figure 1. Network topology used in simulations

Table 1. Parameter configuration in simulations

propagation delay of access link	0 (5.06×10^{-6}), 0.05, 0.1, 0.2	[s]
propagation delay of backbone link	0 (5.06×10^{-6}), 0.05, 0.1, 0.2	[s]
propagation delay of all other links	0 (5.06×10^{-6})	[s]
bandwidth of backbone link	10	[Mbit/s]
bandwidth of all other links	1	[Gbit/s]
MTU of all links	1,500	[byte]
buffer size of backbone router	50	[packet]
buffer size of ingress and egress routers	1	[Mbyte]
the number of TCP tunnels	1	
the number of end-to-end TCP flows	1	
bandwidth of background traffic	3	[Mbit/s]

3 [Mbit/s] and the packet length of 1,500 [byte]. Other parameters of the end-to-end TCP and the tunnel TCP were: the socket buffer size is 640 [Kbyte], the window scale option is enabled, and the buffer size of the ingress and egress routers of the TCP tunnel is 1 [Mbyte]. For all other parameters, initial values of the TCP Reno module in OPNET 10.0A were used. It should be noted that the SACK option is disabled by default in TCP Reno module of OPNET 10.0A.

As performance metrics of the end-to-end TCP, the goodput and the round-trip time of the TCP flown were measured. Samples of the goodput and the round-trip time were measured as the average of 180 [s]. For each parameter configuration, we calculated their average and 95% confidence interval by executing 5 times simulation.

3.2. Effect of propagation delay

In what follows, focusing on the relation between the propagation delay between an end host and the ingress router of the TCP tunnel, and the propagation delay of the TCP tunnel, we evaluate the performance of the end-to-end TCP flow. For this purpose, we performed simulation by changing the propagation delay τ_a of the access link and the propagation delay τ_b of the backbone link. In order to investigate how the end-to-end TCP performance is affected by the TCP tunnel, we also performed simulation without using the TCP tunnel.

First, we focus on the goodput of the TCP flow. Figure 2 shows the goodput of the end-to-end TCP flow in the case of using the TCP tunnel. Moreover, Fig. 3 shows the goodput of the TCP flow when not using the TCP tunnel. These figures show that when using the TCP tunnel, the goodput of the end-to-end TCP decreases slightly as compared with the case wihtout the TCP tunnel. However, when both the propagation delay τ_a of the access link and the propagation delay τ_b of the backbone link are large, using the TCP tunnel increases the goodput of the TCP flow by approximately 1.5 times. The

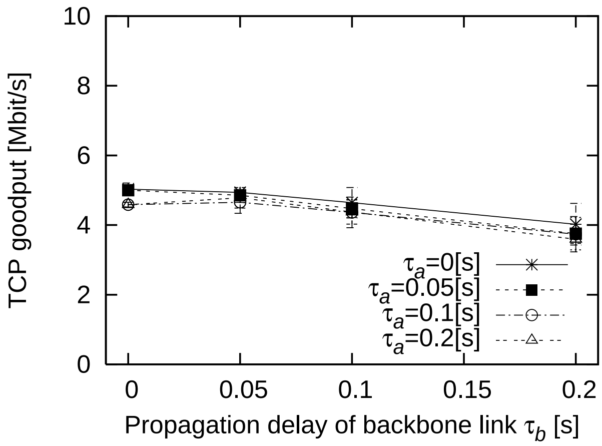

Figure 2. TCP goodput with TCP tunnel (effect of propagation delays of access link and backbone link)

reason that the goodput of the end-to-end TCP flow decreases by using the TCP tunnel in Figs. 2 and 3 can be explained by the phenomenon reported in the literatures.[1, 9] Namely, an end-to-end TCP unnecessarily retransmits packets; for instance, if a packet loss occurs at the backbone link, packet transfer in the tunnel TCP will be delayed. Consequently, a retransmission timeout occurs in the end-to-end TCP, so that the end-to-end TCP retransmits lost packets. However, since lost packets are retransmitted by the tunnel TCP, packets retransmitted by the end-to-end TCP are of no value. Such useless packet retransmissions cause decrease in the goodput of the end-to-end TCP flow.

When the propagation delay of the backbone link and the propagation delay of the access link are large, why using the TCP tunnel increases the goodput of an end-to-end TCP flow can be explained as follows. When a packet loss occurs at the backbone link, not the end-to-end but the tunnel TCP will retransmit the lost packet. Hence, from the end-to-end TCP viewpoint, it seems that there is no packet loss in the network. Therefore, the end-to-end TCP increases its window size, and transmits packets to the ingress router of the TCP tunnel at high speed. On the contrary, the propagation delay of the TCP tunnel is smaller than that of the TCP flow. As the characteristic of TCP, it is known that the goodput of a TCP connection with a larger round-trip time becomes larger than that of a TCP connection with a smaller round-trip time.[7, 8] Hence, using the TCP tunnel increases the goodput of the TCP flow.

Next, we focus on the round-trip time of the end-to-end TCP flow. Figure 4 shows the round-trip time of the TCP flow with the TCP tunnel. Figure. 5 also shows the round-trip time of the TCP flow without the TCP tunnel. These figures show that using the TCP tunnel increases the round-trip time of the TCP flow approximately 1.4 to 3 times.

Packets sent by the end-to-end TCP are once buffered at the ingress router of the TCP tunnel, and are gradually transferred according to the transfer speed of the TCP tunnel. This is the reason of the increase in the round-trip time. Namely, the increase in the round-trip time of the TCP flow is mostly caused by the increased waiting time in the buffer of the ingress router of the TCP tunnel. Usually, TCP is used for applications that can tolerate a certain amount of delay. Hence, even if the round-trip time of the TCP flow increases, it would hardly become a problem in practice.

From these observations, we find that using a TCP tunnel slightly decreases the goodput of a TCP flow except when the propagation delay of an access link and a backbone link are large. We also find that the goodput of TCP flow significantly increases when the propagation delay of an access link and a backbone link are large. Furthermore, we also find that using a TCP tunnel increases the round-trip time of a TCP flow.

3.3. Effect of SACK option

To investigate the effect of the SACK option on the end-to-end TCP performance, we performed simulations when the end-to-end TCP and/or the tunnel TCP uses the SACK option.

Figure 6 shows the goodput of the TCP flow when the TCP tunnel uses the SACK option. Comparison of Figs. 2, 3, and 6 show that the goodput of the end-to-end TCP flow increases when using the TCP tunnel with the SACK option.

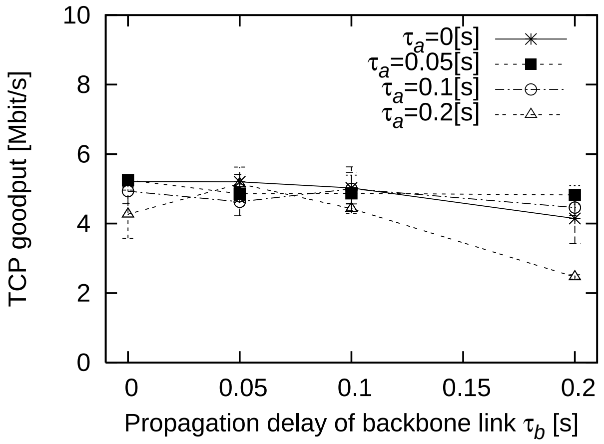

Figure 3. TCP goodput without TCP tunnel (effect of propagation delays of access link and backbone link)

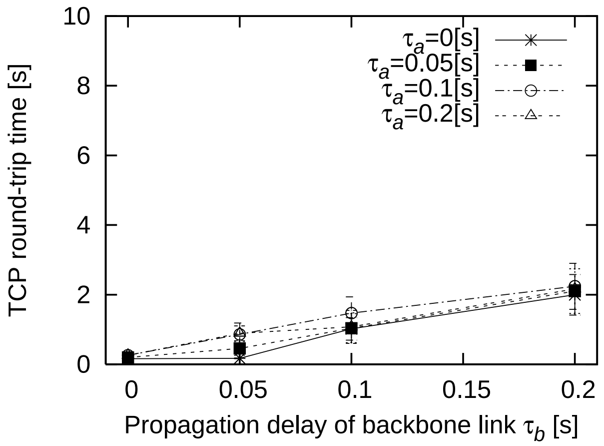

Figure 4. TCP round-trip time with TCP tunnel (effect of propagation delays of access link and backbone link)

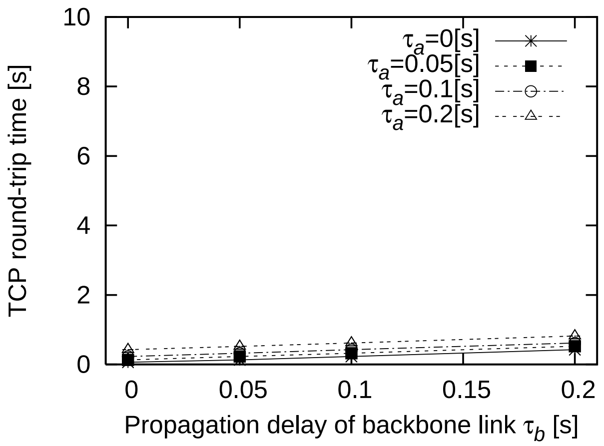

Figure 5. TCP round-trip time without TCP tunnel (effect of propagation delays of access link and backbone link)

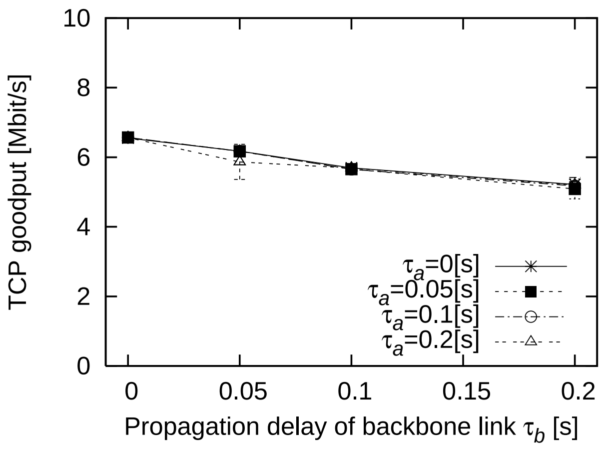

Figure 6. TCP goodput with SACK-enabled TCP tunnel (effect of the SACK option)

Note that, although the simulation result is not included due to space limitation, the goodput of the TCP flow increased when both end-to-end TCP and the tunnel TCP used the SACK option. However, when only thee nd-to-end TCP used SACK option, the goodput of the TCP flow did not increase. The reason that using the tunnel TCP with the SACK option increases the goodput of the TCP flow is probably the same with that reported in the literature. [9] Namely, the tunnel TCP with the SACK option can retransmit lost packets immediately after an ACK packet is received, so that the goodput of the TCP flow is increased.

From these observations, we find that the tunnel TCP with the SACK option improves the goodput of the TCP flow.

3.4. Effect of socket buffer size

In the literature,[9] the authors have shown that using a TCP tunnel increases the goodput of a TCP flow when the socket buffer size of the tunnel TCP is approximately equal to the bandwidth-delay product of a network. However, effect of the socket buffer size of the end-to-end TCP on the end-to-end TCP performance has not been investigated. We therefore performed simulation by changing the socket buffer size of the end-to-end and the tunnel TCP. Specifically, we performed simulation by setting the socket buffer size of either the end-to-end TCP or the tunnel TCP to 32 or 64 [Kbyte].

Figure 7 shows the relation between the propagation delay of the backbone link and the goodput of the end-to-end TCP flow for different socket buffer sizes of the TCP and the tunnel TCP. This figure indicates that the goodput of the TCP flow degrades when the socket buffer size of the end-to-end TCP or the tunnel TCP is small. In particular, when the propagation delay of the backbone link is large, the goodput of the TCP flow degrades significantly. This result implies that it is necessary to configure not only the socket buffer size of the tunnel TCP but also the socket buffer size of the end-to-end TCP according to the bandwidth-delay product of a network.

The round-trip time of the end-to-end TCP flow when changing the socket buffer size of the end-to-end TCP and the tunnel TCP is shown in Fig. 8. The figure shows that the round-trip time of the TCP flow is affected not by the socket buffer size of the tunnel TCP but by the socket buffer size of end-to-end TCP. The decrease in round-trip time can be explained as follows. When the socket buffer size of the end-to-end TCP is small, the amount of traffic that the end-to-end TCP transmits decreases. Consequently, the number of packets kept waiting in the buffer of the ingress router of the TCP tunnel decreases. Thereby, it is thought that the round-trip time of the TCP flow decreases. This is because the great portion of the end-to-end transfer delay is caused by the waiting time in the buffer of the ingress router of the TCP tunnel.

From these observations, we find that it is necessary to configure the socket buffer size of the end-to-end TCP and the socket buffer size of the tunnel TCP according to the bandwidth-delay product of a network.

3.5. Effect of the buffer size of the ingress router of a TCP tunnel

In the above simulations, the buffer size of the ingress router of the TCP tunnel was sufficiently large (i.e., 1 [Mbyte]). Hence, no packet was discarded at the ingress router of the TCP tunnel. However, when the buffer size of the ingress

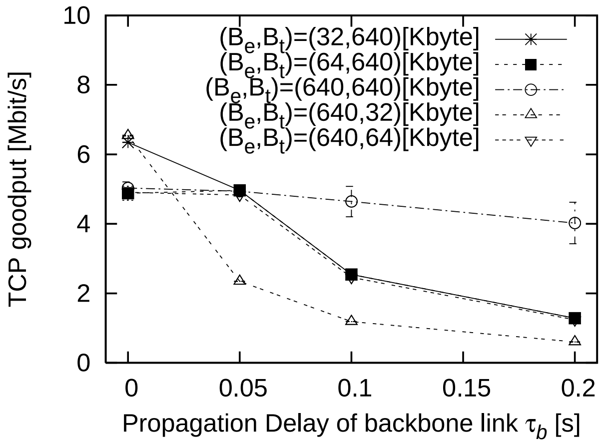

Figure 7. TCP goodput (effect of end-to-end TCP socket buffer size B_e and tunnel TCP socket buffer size B_t)

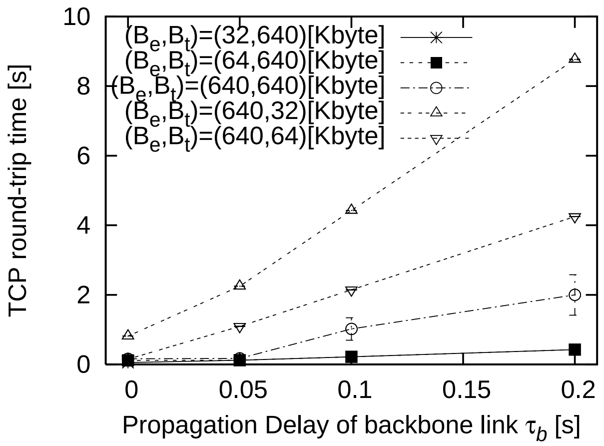

Figure 8. TCP round-trip time (effect of end-to-end TCP socket buffer size B_e and tunnel TCP socket buffer size B_t)

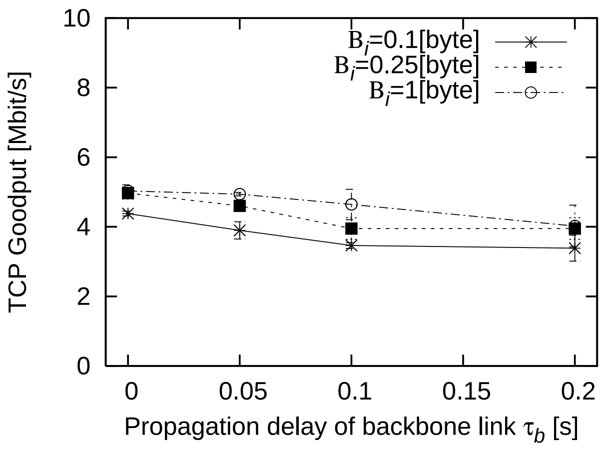

Figure 9. TCP goodput (effect of the ingress router buffer size of TCP tunnel B_i)

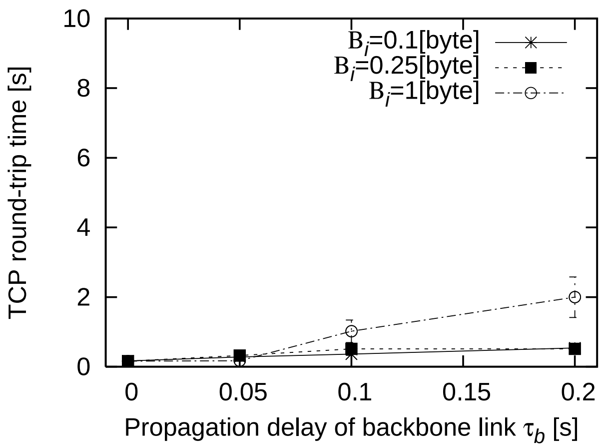

Figure 10. TCP round-trip time (effect of the ingress router sender buffer size of TCP tunnel B_i)

router of the TCP tunnel is small, it is expected that some packets may be discarded at the ingress router of the TCP tunnel. For clarifying effect of the buffer size of the ingress router of the TCP tunnel on the end-to-end TCP performance, we performed simulation while changing the buffer size of the ingress router of the TCP tunnel. Specifically, simulation was repeated by changing the buffer size of the ingress router of the TCP tunnel to 1, 0.25, and 0.1 [Mbyte].

Fig. 9 shows the goodput of the TCP flow when changing the buffer size B_i of the ingress router of the TCP tunnel. This figure shows that the goodput of the TCP flow decreases when the buffer size of the ingress router of the TCP tunnel becomes small. This is because when the buffer size of the ingress router of the TCP tunnel is small, packets are lost at the router and the end-to-end TCP timeouts, so that the goodput of the TCP flow degrades. For instance, when $B_i = 0.25$ and 0.1, packet losses were observed at the ingress router of the TCP tunnel.

Moreover, Fig. 10 shows the round-trip time of the TCP flow when changing the buffer size B_i of the ingress router of the TCP tunnel. This figure shows that the round-trip time decreases when the buffer size of the ingress router of the TCP tunnel becomes small. This is because the buffering time of a packet in the buffer of the ingress router of the TCP tunnel becomes small.

The above observations show that when using a TCP tunnel, the goodput of a TCP flow can be improved by increasing the buffer size of the ingress router of the TCP tunnel at the cost of increase in the round-trip time.

4. CONCLUSIONS AND FUTURE WORKS

In this paper, we have investigated effect of a TCP tunnel on the end-to-end TCP performance, and have shown the desired TCP parameter configuration for improving the end-to-end TCP performance. First, this paper has clearly shown that using a TCP tunnel usually degrades the goodput of the end-to-end TCP flow. However, it has also been found that in the network where the propagation delay is large, the goodput of the end-to-end TCP flow improves. We have shown that using the SACK option solves the problem of the decreased goodput of the end-to-end TCP flow. We have also shown that when the socket buffer size of the end-to-end TCP or the tunnel TCP is not large, the goodput of the end-to-end TCP flow degrades. We have also shown that the buffer size of the ingress router of the TCP tunnel should be large enough for preventing packet losses at the ingress router.

Our future work includes investigating effect of other system parameters and TCP parameters, which have not been taken account of in this paper, on the performance of the end-to-end TCP flow. Specifically, several factors such as the access link bandwidth, the backbone link bandwidth, the number of TCP flows, the number of TCP tunnels, the traffic pattern of TCP flows, version of TCP, and the initial value of RTO, need to be investigated.

REFERENCES

1. B. P. Lee, L. Jacob, W. K. G. Seah, and A. L. Ananda, “Avoiding congestion collapse on the Internet using TCP tunnels,” *Computer Networks* **39**, pp. 207–219, Dec. 2002.
2. H. T. Kung and S. Y. Wang, “TCP trunking: Design, implementation, and performance,” in *Proceedings of IEEE International Conference on Network Protocols ’99*, pp. 222–231, Oct. 1999.
3. “VTun - virtual tunnles over TCP/IP networks.” `http://vtun.sourceforge.net`.
4. M. Jacobson, O. Nordstrom, and R. J. Clark, “HTun: Providing IP service over an HTTP proxy.” `http://htun.runslinux.net/docs/htun-paper.pdf`.
5. “OpenSSH.” `http://www.openssh.com/`.
6. “Virtual ethernet system SoftEther.” `http://www.softether.com/jp/` in Japanese.
7. L. Qiu, Y. Zhang, and S. Keshav, “On individual and aggregate TCP performance,” in *Proceedings of Internetl Conference on Network Protocols*, pp. 203–212, Oct. 1999.
8. S. Floyd, M. Handley, and J. Padhye, “A comparison of equation-based and AIMD congestion control,” tech. rep., ACIRI, 2000. available at `http://www.aciri.org/tfrc/aimd.pdf`.
9. L. Jacob, K. Srijith, H. Duo, and A. Ananda, “Effectiveness of TCP SACK, TCP HACK and TCP trunk over satellite links,” in *Proceedings of IEEE International Conference on Communications*, **5**, pp. 3038–3043, Apr. 2002.
10. E. Altman and K. Avrachenkov, “A stochastic model of TCP/IP with stationary random losses,” in *Proceedings of ACM SIGCOMM 2000*, pp. 231–242, Aug. 2000.
11. M. Mathis, J. Semke, and J. Mahdavi, “The macroscopic behavior of the TCP congestion avoidance algorithm,” *ACM SIGCOMM Communication Review* **27**, pp. 67–82, July 1997.
12. O. Titz, “Why TCP over TCP is a bad idea.” `http://sites.inka.de/sites/bigred/devel/tcp-tcp.html`.
13. Opnet Technologies, Inc., “OPNET.” `http://www.opnet.com/`.