*Engineering*

*Electrical Engineering fields*

Okayama University *Year* 2005

# Performance Improvement of TCP using Performance Enhancing Proxies — Effect of Premature ACK Transmission Timing on Throughput —

Shigeyuki Osada* Tokumi Yokohira† Wang Hui‡
Kiyohiko Okayama** Nariyoshi Yamai††

*Okayama University
†Okayama University
‡Henan University of Science and Technology
**Okayama University
††Okayama University

# Performance Improvement of TCP using Performance Enhancing Proxies
## — Effect of Premature ACK Transmission Timing on Throughput —

Shigeyuki Osada[†], Tokumi Yokohira[†], Wang Hui[††], Kiyohiko Okayama[†††] and Nariyoshi Yamai[†††]
[†]The Graduate School of Natural Science and Technology, Okayama Univ.,
Tsushimanaka 3-1-1, Okayama, 700-8530, Japan
[††]The Faculty of Electric Information Engineering, Henan Univ. of Science and Technology, LuoYang, China
[†††]The Information Technology Center, Okayama Univ., Okayama, Japan
E-mail : osada@net.cne.okayama-u.ac.jp, yokohira@cne.okayama-u.ac.jp, wh@mail.haust.edu.cn,
okayama@cc.okayama-u.ac.jp, yamai@cc.okayama-u.ac.jp

## Abstract

In order to improve TCP performance, a method using a PEP (Performance Enhancing Proxy) is proposed. The PEP operates on a router along a TCP connection. When a data packet arrives at the PEP, it forwards the packet to the destination host, transmits the corresponding ACK (premature ACK) to the source host in behalf of the destination host and stores the copy of the packet into its own buffer (PEP buffer) in case of the retransmission of the packet. In this paper, under the strategy which keeps the number of packets in the PEP buffer for which premature ACKs have been returned being less than or equal to a fixed threshold value (watermark value), we investigate the relation between the watermark value and the maximum throughput. Extensive simulation runs show that the simulation results are roughly classified into two cases. One case is that the maximum throughput becomes larger for larger watermark value and becomes a constant value when the watermark value is over a value. The other case is that though the maximum throughput becomes larger for lager watermark value in the same way, it reversely decreases when the watermark value is over a value. We also show that the latter (former) case is easier to occur as the propagation delay in the input side network of the PEP becomes smaller (larger) and the propagation delay in the output side network of the PEP becomes larger (smaller) and the PEP buffer capacity becomes smaller (larger).

**Key words** TCP, PEP, Premature ACK, watermark

## 1 Introduction

Transmission Control Protocol (TCP) [1] performs very well in networks with small delays. However, in networks with large delays, the performance of TCP is poor [2]-[5]. That is, in networks with large delays, the slow-start duration increases because the congestion window does not increase rapidly due to large delays and the sender must often wait for open of the window. In order to mitigate these problems without modifications of TCP specification, deploying a *Performance Enhancing Proxy (PEP)* has been proposed [6].

The PEP operates on a router along a TCP connection. It monitors all the packets of a TCP connection through it. When it receives a data packet from the source host, it transmits the packet to the destination host, copies the packet into its own buffer (*PEP buffer*) in case of the retransmission of the packet, and sends an ACK (*premature ACK*) acknowledging the packet to the source host instead of the destination host. When the PEP receives an ACK (*real ACK*) from the destination host, it removes the packets from the PEP buffer which the real ACK acknowledges, and drops the real ACK. When a real ACK for a packet for which a premature ACK has been returned does not arrive at the PEP from the destination host during a timeout value, the PEP retransmits the packet's copy to the destination host.

While generating premature ACKs in the PEP improves TCP performance, many premature ACKs may cause the congestion collapse because they let the source host send a large number of data packets. Thus, it is necessary for the PEP to introduce a criterion which decides whether to send premature ACKs to the destination host or not. The paper [7] introduces a criterion in which a *watermark* is used. The criterion is one that keeps the number of packets in the PEP buffer for which premature ACKs have been returned being less than or equal to a fixed threshold value (the watermark value). Through the paper has showed that the criterion improves TCP performance, the relation between the watermark value and TCP throughput is not sufficiently investigated.

The goal of this paper is to research the relation for various values of propagation delay and the PEP buffer capacity. We introduce a network model where there is a network with only two links, and the source host and destination host are placed at the ends of the respective

links and the PEP is placed at the joining point of the two links. Incorporating the network model into the NS2 simulator [8], we obtain the maximum throughput for various values of the propagation delays of the two links, the PEP buffer capacity and the watermark, and discuss the relation between the watermark value and the maximum throughput.

The remainder of this paper is organized as follows. Section 2 explains the behavior of the PEP in detail. Section 3 describes the simulation model, and Section 4 discusses the relation between the watermark value and the maximum throughput.

# 2 Behavior of PEP

## 2.1 Network Environment

The PEP operates on an edge router connecting different networks. Figure 1 shows network environment where the PEP is used. S, P and D are a source host, a PEP router and a destination host, respectively. The source host and the PEP communicate with each other via a network (S-Net), and the PEP and the destination host communicate with each other via a network (D-Net). For the simplicity of the description, we assume that the destination host does not send data packets to the source host, and assume that only one (IP) data packet is generated from each TCP segment and the length of each data packet is equal to each other, and an integer sequence number is assigned to each packet in its generation order.

## 2.2 PEP Behavior

The PEP in this paper is based on TCP ACK manipulation. It monitors all the packets of a TCP connection through it and acknowledges some packets instead of the destination host. In order to perform the ACK manipulation, the PEP has two kinds of buffers, a normal FIFO store-and-forward (SF) buffer and a special FIFO buffer called PEP buffer, as shown Fig. 2. We assume that there is only one SF buffer for each output link and one PEP buffer for each TCP connection.

In the following subsections 2.2.1 and 2.2.2, we describe the PEP behavior when it receives a data packet and an ACK packet, respectively.

### 2.2.1 Behavior When a Data Packet is Received

When the PEP receives a data packet DP from the source host, it executes one of the followings (A-1) ~ (A-3), where let $N$ and $n$ denote the PEP buffer capacity and the number of data packets (copies) in the PEP buffer when DP is received, and let $W$ ($0 \leq W \leq N$) and $w$ ($0 \leq w \leq W$) denote the fixed initial value of the watermark and the current value of the watermark, respectively.

(A-1) In the case that $w < n$: The PEP stores DP into the SF buffer to forward DP to the destination host. And it stores DP's copy into the PEP buffer, marks it with "p-acked", and returns the corresponding premature ACK to the source host.

(A-2) In the case that $N > n \geq w$: The PEP stores DP into the SF buffer to forward DP to the destination host. And it stores DP's copy into the PEP buffer and marks it with "stored". Note that the PEP does not return a premature ACK.

(A-3) In the case that $n = N$: The PEP does nothing except for storing DP into the SF buffer to forward DP to the destination host. We call such DP a *through packet*. □

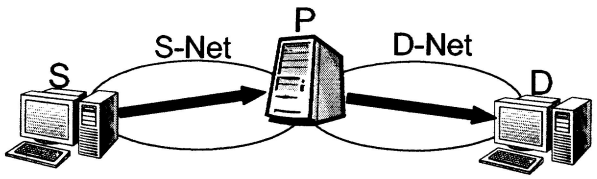


Figure 1: Network environment.

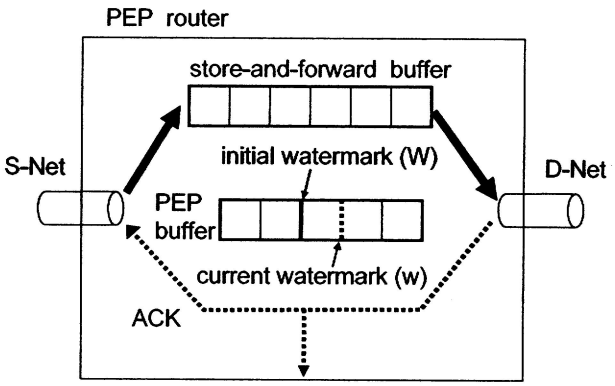


Figure 2: PEP model.

### 2.2.2 Behavior When an Acknowledged Packet is Received

When a real ACK from the destination host is received, the PEP executes the following steps (B-1) ~ (B-5) in this order, where in the case that there exists an unacknowledged through packet when the real ACK is received, let $n_{min}$ denote the minimum value among the sequence numbers of all the unacknowledged through packets.

(B-1) The PEP removes all the packets from the PEP buffer which the real ACK acknowledges.

(B-2) The PEP shifts all the packets located at left side of the removed packets to the right direction.

(B-3) The PEP sets the value of $w$ according to one of the followings (B-3-1) and (B-3-2).

(B-3-1) In the case that there does not exist an unacknowledged through packet, or in the case that there exists an unacknowledged through packet

and there does not exist a packet whose sequence number is greater than $n_{min}$ in the first $W$ positions of the PEP buffer: The PEP sets $w = W$.

(B-3-2) In the case that there exists an unacknowledged through packet and there exists a packet whose sequence number is greater than $n_{min}$ in the first $W$ positions of the PEP buffer: The PEP finds a packet whose sequence number is maximum among the sequence numbers of all the packets whose sequence numbers are less than $n_{min}$, and sets the value of $w$ to the left position of the found packet.

(B-4) If the real ACK acknowledges an unacknowledged through packet and/or a packet marked with "stored", the PEP returns the real ACK to the source host, otherwise it simply discards the real ACK because the corresponding premature ACK has been already returned.

(B-5) If there exists a packet marked with "stored" in the first $w$ positions of the PEP buffer, then the PEP returns the corresponding premature ACK to the source host, and marks the packet with "p-acked". □

The step (B-3) described above guarantees that premature ACKs are returned for only the packets located at the right positions of the current watermark value $w$. For example, assume that there are currently twelve packets in the PEP buffer as shown in Fig. 3(a), where the packets 15 and 22 are unacknowledged through packets, and consequently $n_{min} = 15$. When a real ACK acknowledging the packets 10 ~ 12 is received, the steps described above change the PEP buffer from Fig. 3(a) to Fig. 3(b). Note that $w$ is at the left position of the packet 14 using (B-3-2). And note that if the value of $w$ is not adjusted, that is, $w$ remains unchanged ($w = 4$), because the packets 16 and 17 are in the right position of the current watermark value, a premature ACK for the packets 16 and 17 is returned, and consequently the packet 15 is also acknowledged due to the acknowledge accumulation property of TCP, which is a misbehavior. Next, we assume that a real ACK acknowledging the packets 13 ~ 16 is returned. The PEP changes the PEP buffer from Fig. 3(b) to Fig. 3(c). Note that when the real ACK is returned, because $n_{min}$ is changed from 15 to 23 and the sequence numbers 17 ~ 20 of the packets located at the first $W$ positions are less than $n_{min}$, the step (B-3-1) sets $w = W$.

When a real ACK for a packet for which a premature ACK has been returned does not arrive at the the PEP from the destination host during a timeout value, the PEP considers that the corresponding packet which has been already transmitted is lost, and consequently it retransmits the packet's copy to the destination host. We omit the description of the retransmission procedure, because we assume in this paper that there is no packet loss in the networks.

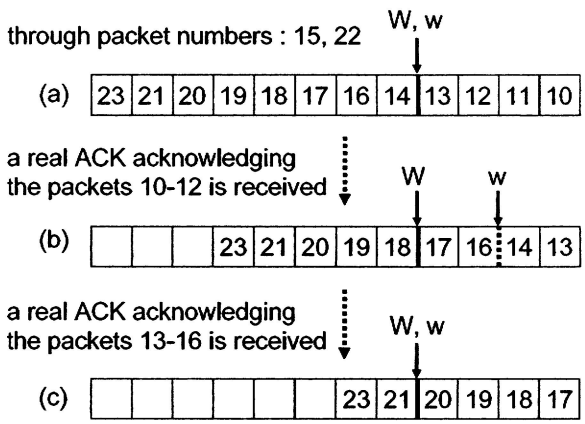


Figure 3: Adjustment of the current watermark value $w$.

# 3 Simulation Model

## 3.1 Network Model

In this paper, we assume that each of S-Net and D-Net has only one link, and assume that there is no packet loss and no bit error in the links. It is assumed that there exist infinite number of packets to be transmitted in the source host and the source host tries to transmit the packets according to the TCP New Reno protocol [5] with the initial congestion window size of one packet and assumed that the destination host always advertises the window size of 20 packets via real ACKs and also PEP always advertises the window size of 20 packets via premature ACKs. The packet length is assumed to be 1000 bytes.

## 3.2 Simulation Parameters

After incorporating the PEP function into the NS2 simulator, we simulated for every combination of the propagation delay $d_S$ and transmission capacity $v_S$ of S-Net, the propagation delay $d_D$ and transmission capacity $v_D$ of D-Net, the PEP buffer capacity $N$ and the initial watermark value $W$ (simply watermark hereafter) shown in Table 1, and obtained throughput for every combination, where throughput is defined as the total number of the bits of the all packets which have arrived at the destination host in every one second interval. When $W = 0$, the value of throughput means the value of throughput in the original TCP without the PEP function.

# 4 Effect of the Watermark Value on the Maximum Throughput

We can roughly classify the simulation results into two cases with respect to the relation between the initial watermark value and the maximum throughput.

Table 1: Simulation parameters.

| | |
|---|---|
| propagation delay of S-Net $d_S$ [ms] | 1, 10, 30, 50, 70, 80, 90, 100, 120, 130, 150, 200 |
| transmission speed of S-Net $v_S$ [Mbps] | 10 |
| propagation delay of D-Net $d_D$ [ms] | 10, 30, 50, 80, 100, 120, 130, 150, 200 |
| transmission speed of D-Net $v_D$ [Mbps] | 1 |
| PEP buffer capacity $N$ [packets] | 30, 50, 70, 100 |
| Initial Watermark $W$ [packets] | $0 \sim N$ |

(R-1) The maximum throughput becomes larger for larger watermark value and becomes a constant value when the watermark value is over a value.

(R-2) Though the maximum throughput becomes larger for lager watermark value in the same way as (R-1), it reversely decreases when the watermark value is over a value.

We describe the two cases (R-1) and (R-2) in detail in the following subsections 4.1 and 4.2, respectively, and describe the relation between the cases and the network parameters $d_S$, $d_D$ and $N$ (note that because the values $v_S$ and $v_D$ are fixed at 10Mbps and 1Mbps, respectively as shown in Table 1, we does not consider $v_S$ and $v_D$ as network parameters in this paper).

## 4.1 In the Case that the Maximum Throughput does not Decrease When the Watermark Value is Large

Figure 4 shows an example of the case (R-1), where $d_S$ = 80ms, $d_D$ = 100ms, and $N$ = 50 packets. We can see that slow-start period shortens and the maximum throughput increases when the value is less than 25. This is because the number of premature ACKs becomes larger for larger watermark value, and consequently the source host can open its congestion window more quickly and transmit more packets. When the watermark value is greater than or equal to 25, the maximum throughput reaches the maximum available bandwidth which is equal to the transmission speed $v_D$ of D-Net (1Mbps). This is because the source host can transmit packets in the state that TCP window always open, and consequently it can make full use of the maximum available bandwidth.

Figure 5 shows another example of the case (R-1), where $d_S$ = 100ms, $d_D$ = 100ms, and $N$ = 50 packets. In this figure, in the same way as Fig. 4, though the maximum throughput increases as the watermark value increases, and reaches a maximum value (about 0.8Mbps), it can not reach the maximum available bandwidth (1Mbps). This is because the propagation delay $d_S$ of S-Net is large. Even if the watermark value increases it takes much time for the source host to receive premature ACKs from the PEP due to the large propagation delay of S-Net. Thus, the packet sending rate of the source host is limited by its advertised window size (20 packets) (the source host must frequently wait for open of the TCP window), and consequently the source host can not use the maximum available bandwidth fully. In order to improve such situation, we can consider some strategies. One is to use multiple PEPs along the route of a TCP connection. Using multiple PEPs, we can shorten the propagation delay time between the source host and the first PEP located on the route. Another strategy is that the PEP announces larger window size than that of the destination host. Research on performance improvement using such strategies is one of the future works.

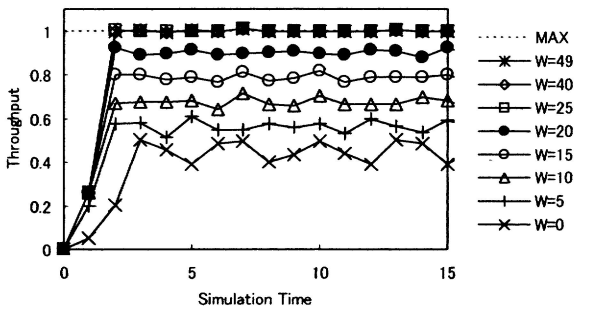


Figure 4: The relation between the watermark value and throughput ($d_S$ = 80, $d_D$ = 100, $N$ = 50).

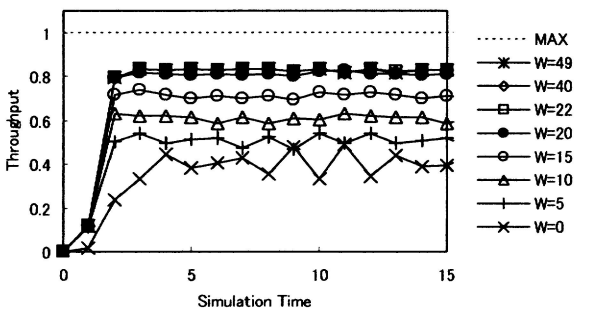


Figure 5: The relation between the watermark value and throughput ($d_S$ = 100, $d_D$ = 100, $N$ = 50).

## 4.2 In the Case that the Maximum Throughput Decreases When the Watermark Value is Large

Figure 6 shows an example of the case (R-2), where $d_S$ = 50ms, $d_D$ = 100ms, and $N$ = 50 packets. We can see that slow-start period shortens and the maximum throughput increases as the watermark value increases in the same way as the situations in 4.1 when the watermark value is less than or equal to 38. However, it is differ-

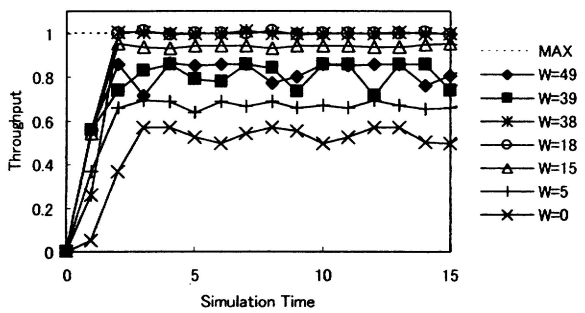


Figure 6: The relation between the watermark value and throughput ($d_S$ = 50, $d_D$ = 100, $N$ = 50).

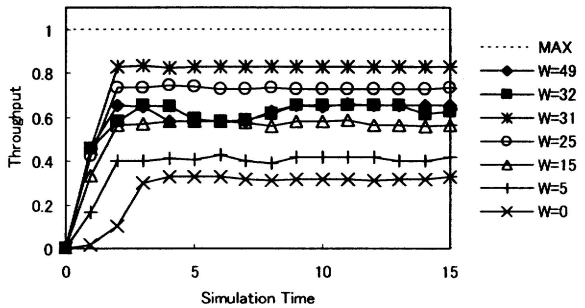


Figure 7: The relation between the watermark value and throughput ($d_S$ = 50, $d_D$ = 200, $N$ = 50).

ent from the situations above that the maximum throughput decreases when the watermark value is large (the watermark is greater than or equal to 39). This is because that through packets occur due to the overflow of the PEP buffer. When there is an unacknowledged through packet, the PEP can not transmit premature ACKs until it receives the real ACK acknowledging the unacknowledged through packets. For example, assume that there are currently eight packets in the PEP buffer as shown in Fig. 8(a), where the packet 16 is an unacknowledged through packet. When a real ACK acknowledging the packet 10 is received, the steps described in the section 2.2.2 change the PEP buffer from Fig. 8(a) to Fig. 8(b), and the PEP sends a premature ACK acknowledging the packet 15 marked "stored". However, when a real ACK acknowledging the packet 11 is received, the value of $w$ is adjusted so that the PEP does not acknowledge the unacknowledged through packet 16, and the PEP can not send a premature ACK as shown in Fig. 8(c). Therefore the source host can not receive premature ACKs from the PEP as long as there is an unacknowledged through packet, and the source host can not send data packets any more.

Figure 7 shows another example of the case (R-2), where $d_S$ = 50ms, $d_D$ = 200ms, and $N$ = 50 packets. We can see that slow-start period shortens and the maximum throughput increases as the watermark value increases in the same way as the situations 4.1 when the

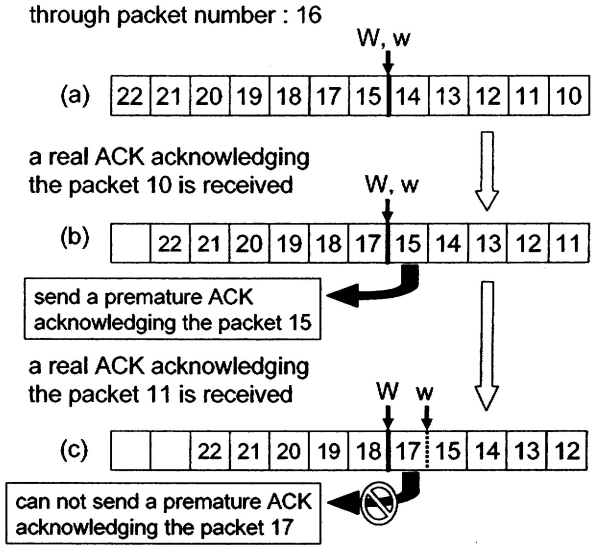


Figure 8: The reason that the maximum throughput decreases when there is an unacknowledged through packet.

watermark value is less than or equal to 31. However, in this figure we can see that the maximum throughput can not reach the maximum available bandwidth (1Mbps) and it decreases when the watermark value is large (when the watermark value is greater than or equal to 32). The reason is trivial merging by the reasons of Figs. 4 and 5.

## 4.3 Relation Between the Change of the Maximum Throughput and the Network Parameters

In this subsection, we discuss which case of the cases (R-1) and (R-2) is easier to occur according to the values of the network parameters $d_S$, $d_D$ and $N$. As described in Section 4.2, the maximum throughput decreases when through packets occur. Thus, the maximum throughput is easier to decrease as the PEP buffer is easier to overflow. The overflow occurrence rate obviously depends on the packet arrival and departure rates of the PEP buffer and its capacity.

The smaller value of $d_S$ is, the larger the packet arrival rate is, because premature ACKs can return more quickly and consequently packets from the source host can arrive at the PEP more quickly for the smaller value of $d_S$. Thus, the maximum throughput is easier to decrease for smaller value of $d_S$. This is verified by comparing Figs. 4 and 6 where only the values of $d_S$ are different from each other. While the maximum throughput in Fig. 4 with larger value of $d_S$ does not decrease, the maximum throughput in Fig. 6 with smaller value of $d_S$ decreases.

Regarding the packet departure, a similar discussion holds. The larger the value of $d_D$ is, the smaller the packet departure rate is, because packets from the PEP arrives at the destination host more slowly and consequently real

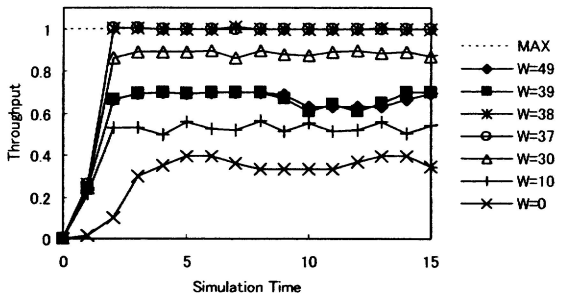


Figure 9: The relation between the watermark value and throughput ($d_S$ = 80, $d_D$ = 150, $N$ = 50).

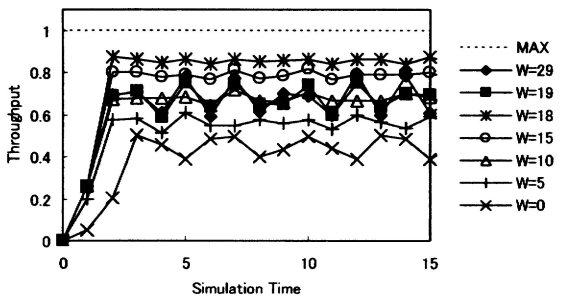


Figure 10: The relation between the watermark value and throughput ($d_S$ = 80, $d_D$ = 100, $N$ = 30).

ACKs from the destination host arrives at the PEP more slowly for the larger value of $d_D$. Thus, the maximum throughput is easier to decrease for larger value of $d_D$. In order to verify this, we show the simulation results for $d_S$ = 80ms, $d_D$ = 150ms ans $N$ = 50 packets in Fig. 9. In Figs. 4 and 9, only the values of $d_D$ are different from each other. While the maximum throughput in Fig. 4 with smaller values of $d_D$ does not decrease, the maximum throughput in Fig. 9 with larger values of $d_D$ decreases.

Finally, we describe the effect of the PEP buffer capacity. The overflow is obviously easier to occur for smaller value of the PEP buffer capacity. Thus, the maximum throughput is easier to decrease for smaller value. Figure 10 show the simulation results for $d_S$ = 80ms, $d_D$ = 100ms and $N$ = 30 packets. In Figs. 4 and 10, only the values of $N$ are different from each other. While the maximum throughput in Fig. 4 with larger value of $N$ does not decrease, the maximum throughput in Fig. 10 with smaller value of $N$ decreases.

# 5 Conclusions

In this paper, we focused on using a PEP to improve the TCP performance, and we investigated the effect of the watermark value, which is used to determine whether the PEP sends a premature ACK or not, on the maximum throughput.

Extensive simulation runs showed that the simulation results are roughly classified into two cases. One case is that the maximum throughput becomes larger for larger watermark value and becomes a constant value when the watermark value is over a value. The other case is that though the maximum throughput becomes larger for lager watermark value in the same way, it reversely decreases when the watermark value is over a value. We also show that the latter (former) case is easier to occur as the propagation delay in the input side network of the PEP becomes smaller (larger) and the propagation delay in the output side network of the PEP becomes larger (smaller) and the PEP buffer capacity becomes smaller (larger).

One of our future works is to investigate how to select the watermark value adequately. As described in 4.1, performance improvement using multiple PEPs and larger window size announcement from the PEP is also included in our future works.

# References

[1] J. Postel, "Transmission Control Protocol," RFC 793, Sep. 1981.

[2] W. Stevens, *TCP/IP Illustrated, Volume 1: The Protocols*. Addison-Wesley, 1994.

[3] M. Allman, V. Paxson, W. Stevens, "TCP Congestion Control," RFC 2581, Apr. 1999.

[4] M. Allman et al., "Ongoing TCP Research Related to Satellites," RFC 2760, Feb. 2000.

[5] S. Floyd, T. Henderson and A. Gurtov, "The NewReno Modification to TCP's Fast Recovery Algorithm," RFC 3782, Apr. 2004.

[6] J. Border et al., "Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations," RFC 3135, Feb. 2001.

[7] D. Dutta and Y. Zhang, "An Active Proxy Based Architecture for TCP in Heterogeneous Variable Bandwidth Network," IEEE GLOBECOM2001, pp. 2316-2320, Nov. 2001.

[8] The Network Simulator - NS2, http://www.isi.edu/nsnam/ns/