TCP Congestion Control Method of Improving Friendliness over Satellite Internet

Hiroyasu Obata, Shingo Nishimoto, and Kenji Ishida
Hiroshima City University
Graduate School of Information Sciences
Hiroshima, Japan
{obata@, nishimoto@net.info., ishida@}hiroshima-cu.ac.jp

***Abstract*—The number of wireless Internet service users that use wireless LAN or satellite links has increased. The broadband satellite Internet services have especially attracted attention because of its characteristics such as global coverage and robustness. TCP (Transmission Control Protocol) is used by many typical applications over satellite Internet. However, a typical TCP (such as TCP-NewReno) which has been developed for wired networks performs poorly in wireless networks. Furthermore, the long propagation delay of satellite links decreases performance of the typical TCP. TCP-STAR has been proposed to solve these problems by modifying TCP congestion control method. TCP-STAR achieves high-speed communication by using the estimated bandwidth. However, if TCP-STAR coexists with the typical TCP, TCP-STAR tends to reduce throughput of the typical TCP. On the other hand, TCP-Fusion has been proposed for wired high-speed networks. TCP-Fusion which uses delay-based and loss-based congestion control method achieves scalability and friendliness to the typical TCP. However, TCP-Fusion cannot obtain high performance over satellite Internet, since TCP-Fusion is developed for wired high-speed links. In this paper, we propose a TCP congestion control method for improving friendliness over satellite links. The proposed method combines TCP-Fusion's congestion control method and TCP-STAR's congestion control method. We evaluate the performance of the proposed method over satellite Internet using ns2 (Network Simulator version 2). As a result, the proposed method achieves good friendliness when the proposed method coexists with the typical TCP. Furthermore, it is found that the proposed method utilizes bandwidth of satellite links well.**

***Index Terms*—Satellite Internet, TCP Congestion Control, Friendliness**

I. Introduction

Recently, bandwidth of wireless links becomes larger by the progress of wireless communication technology. Especially, the satellite link speed will be improved like optical networks. For example, WINDS (Wideband InterNetworking engineering test and Demonstration Satellite)[1][2] (called Kizuna) is developed by JAXA (Japan Aerospace Exploration Agency) and NICT (National Institute of Information and Communications Technology) in Japan for a gigabit satellite communications. WINDS is a geostationary satellite (which covers Asia/Pacific region) and it can communicate interactively at a rate of up to 1.2Gbps. WINDS is now verified its performance by JAXA, NICT, and other organizations. The some authors are consistent members of WINDS experiments organization[3]. Thus, WINDS can contribute a wide area broadband communication.

TCP (Transmission Control Protocol)[4] is used by many typical applications over satellite Internet. However, a typical TCP (TCP-NewReno) which has been developed for wired networks performs poorly in wireless networks due to its sporadic high bit error rate. Furthermore, the long propagation delay of satellite links decreases performance of the typical TCP. TCP-STAR[5][6][7] has been proposed to solve these problems by modifying TCP congestion control method. TCP-STAR achieves high-speed communication by using the estimated bandwidth. TCP-STAR only needs modification of sender side TCP. However, if TCP-STAR coexists with the typical TCP, TCP-STAR tends to reduce throughput of the typical TCP. On the other hand, TCP-Fusion[8] has been proposed for wired high-speed networks. TCP-Fusion which uses delay-based and loss-based congestion control method achieves scalability and friendliness to the typical TCP. However, TCP-Fusion cannot obtain high performance over satellite Internet, since TCP-Fusion is developed for not satellite links but wired high-speed links[9].

In this paper, we propose a TCP congestion control method for improving friendliness over satellite links. The proposed method combines TCP-Fusion's congestion control method and TCP-STAR's congestion control method. We evaluate the performance of the proposed method over satellite Internet using ns2 (Network Simulator version 2)[10]. As a result, the proposed method achieves good friendliness when the proposed method coexists with the typical TCP. Furthermore, it is found that the proposed method utilizes bandwidth of satellite links well.

The rest of this paper is constructed as follows. We describe the related works in Section II. In Section III, we explain the proposed method. Section IV includes evaluations and discussions based on the results from simulation experiments. We summarize this paper at Section V.

II. Related Works

Our work was partially inspired by TCP-STAR for satellite Internet and TCP-Fusion for high speed networks. This section describes TCP-STAR and TCP-Fusion.

A. TCP-STAR

TCP-STAR[5][6][7] has been proposed to improve the throughput over satellite Internet. TCP-STAR is the congestion control method which consists of three mechanisms; Congestion Window Setting (CWS) based on available bandwidth,

978–1–4244–4657–5/09/$25.00 © 2009 IEEE ICICS 2009

Lift Window Control (LWC), and Acknowledgment Error Notification (AEN). In CWS and LWC, TCP-STAR uses ABE (Available Bandwidth Estimation) in TCP-J[11][12] as the available bandwidth estimation method.

1) CWS (Congestion Window Setting): CWS avoids the reduction of the transmission rate using estimated bandwidth when data losses are caused by bit error.

2) LWC (Lift Window Control): LWC increases the congestion window size quickly using both the window control of TCP-Reno and the values based on the results of ABE. Thus, the congestion window becomes larger quickly over long fat networks.

3) AEN (Acknowledgement Error Notification): In satellite networks, the retransmission often occurs by ack losses or delay, because satellite links have large RTT[17]. As a result, the unnecessary timeout and mis-retransmission occurs. TCP-STAR is able to avoid the reduction of throughput by mis-retransmission of data by using AEN.

TCP-STAR can improve the throughput of TCP in the satellite line. And TCP-STAR only requires modification of the sender side TCP module in the terminal. However, the change of all the sending terminals is not easy. Therefore, there is possibility that the modified terminal exists together to terminals of the unmodified. In the environment, TCP-STAR will press typical TCP flows that coexist together with TCP-STAR flows. TCP-STAR has the possibility of depressing the throughput that the typical TCP originally obtained.

B. TCP-Fusion

In order to satisfy efficiency and friendliness tradeoff of TCP, TCP-Fusion has an approach combining a loss-based protocol and delay-based protocol. The key concept of TCP-Fusion is that congestion window sizes are increased aggressively whenever the network is estimated underutilized. On the other hand, when the network is fully utilized, TCP-Fusion tends to perform like the typical TCP. Therefore, TCP-Fusion has a special feature which tries to utilize the residual capacity effectively without impacts on coexisting typical TCP flows.

The congestion control method of TCP-Fusion is shown as follows.

1) Congestion Window Decrement: TCP-Fusion adopts optimization of the decrease parameter based on TCP Westwood-RE (Rate Estimation)[13][14] to improve efficiency particularly in the leaky pipepipe, i.e., the link with large bandwidth-delay product and non-negligible random losses.

In TCP Westwood-RE, the decrease parameter after a loss can be expressed as RTTmin/RTT, where RTTmin and RTT are the minimum RTT and the RTT right before the packet loss, respectively.

2) Congestion Window Increment: Similar to TCP-Vegas, TCP-Fusion has three phases; Increment phase, decrement phase, and steady phase, which are switched by a number of packets in the bottleneck queue.

In wired networks, TCP-Fusion keeps compatibility with a typical TCP solving the scalability problem of TCP. However, TCP-Fusion cannot get an enough performance in the satellite Internet due to its high bit error rate and long propagation delay.

III. PROPOSED METHOD

In order to overcome the problems of existing methods, we propose a new TCP congestion control method which can obtain good friendliness for the typical TCP. The proposed method is obtained by combining TCP-Fusion and CWS/LWC of TCP-STAR. It is assumed that the proposal will be able to obtain good friendliness by using TCP-Fusion and higher throughput by CWS and LWC over the satellite Internet. Following subsections show the detail of the proposed method.

A. Congestion Window Decrement

When the proposal detects packet losses, the congestion window ($cwnd$) and slow start threshold ($ssthresh$) are set by using CWS.

1) Detection of Packet Losses by Duplicate ACKs: If the proposed method detects packet losses by duplicate ACKs, it sets $cwnd$ and $ssthresh$ by Eq.(1).

$$\begin{aligned}
&ssthresh = max(BW_{RE} \times \frac{RTT_{min}}{packet_size}, \frac{cwnd_{last}}{2}) \\
&\quad \text{if } (cwnd_{last} < ssthresh) \\
&\quad\quad cwnd_{new} \Longrightarrow \text{keep recent value} \\
&\quad \text{else if } (cwnd_{last} \geq ssthresh) \\
&\quad\quad cwnd_{new} = ssthresh \qquad (1)
\end{aligned}$$

In Eq.(1), $cwnd_{new}$, $cwnd_{last}$, BW_{RE}, RTT_{min}, $packet\_size$ indicate updated congestion window, previous congestion window, estimated available bandwidth, minimum round trip time, and packet size, respectively. BW_{RE} is obtained by using Rate Estimation (RE) which is one of the mechanism of TCP-Westwood[13][14].

2) Detection of Packet Losses by Retransmission Timeout: If the retransmission timeout occurs, the proposed method sets $cwnd$ and $ssthresh$ by Eq.(2).

$$\begin{aligned}
&ssthresh = max(BW_{RE} \times \frac{RTT_{min}}{packet_size}, \frac{cwnd_{last}}{2}) \\
&cwnd_{new} = 1 \qquad (2)
\end{aligned}$$

B. Congestion Window Increment

Original TCP-Fusion has three phases (increment phase, decrement phase, and steady phase) in case of updating the congestion window. In the proposed method, we apply LWC of TCP-STAR in the increment phase of TCP-Fusion. Window control of decrement phase and steady phase are same as original TCP-Fusion. Eq.(3) shows the congestion window behavior of the proposed method which uses LWC.

$$\begin{aligned}
&cwnd_{new} = \begin{cases} cwnd_{last} + \frac{target_win}{cwnd_{last}}, \text{ if } \; diff < \alpha \\ cwnd_{last} + \frac{(-diff+\alpha)}{cwnd_{last}}, \text{ if } \; diff > 3 \times \alpha \\ cwnd_{last}, \text{otherwise} \end{cases} \\
&cwnd_{new} = cwnd_{reno}, \text{ if } \; cwnd_{new} < cwnd_{reno} \qquad (3)
\end{aligned}$$

TABLE I
SIMULATION PARAMETERS.

Packet size [byte]	1500
Buffer size of router [Kbyte]	6250
Simulation time [s]	100

In Eq.(3), $diff$, α, $target\_win$ indicate the number of estimated packets in bottleneck router queue, lower bound threshold to switch three phases, and additional window size by LWC, respectively. $target\_win$ is calculated by Eq.(4).

$$target\_win = cwnd_{reno} + cwnd_{abe} - cwnd_{last}$$
$$cwnd_{abe} = BW_{ABE} \times \frac{RTT_{min}}{packet\_size} \quad (4)$$

In Eq.(4), BW_{ABE} shows the bottleneck bandwidth and it is obtained by using ABE of TCP-STAR.

IV. SIMULATION EXPERIMENTS

This paper evaluates the proposed method by using network simulator ns2[10] and compares the proposed method with TCP-STAR and TCP-Fusion.

A. Simulation Settings

Fig.1 and Table.I show a network topology and network parameters, respectively. This paper sets a link between Router1 and Router2 as satellite link, and random packet losses occurs in the satellite link. The supposed application in this simulation is FTP which has N flows. In simulation, $N/2$ typical TCP flows (TCP-NewReno) shares the bottleneck satelllite link with $N/2$ proposed method (or TCP-STAR or TCP-Fusion) flows. All simulation results are obtained by averaging ten trials. All TCP variants use window scale option[15] and SACK option[16]. And the buffer size of Router1 and Router2 is set equal to bandwidth delay product[17].

This paper evaluates friendliness and bandwidth utilization. We uses a throughput ratio (Tr) in order to evaluate friendliness of the proposed method. Tr is obtained by Eq.(5)

$$Tr = \frac{T_{mixed}}{T_{allreno}} \quad (5)$$

In Eq.(5), T_{mixed} indicate the average throughput of typical TCP (TCP-NewReno) flows when the proposed method coexists. And $T_{allreno}$ shows the average throughput of typical TCP (TCP-NewReno) flows when all flows are TCP-NewReno. Thus, when Tr gets close to 1, it indicates the friendliness is high. The bandwidth utilization U is calculated by Eq.(6). In Eq.(6), T_{total} and C indicate the total throughput of all flows and the bottleneck link bandwidth, respectively.

$$U = \frac{T_{total}}{C} \quad (6)$$

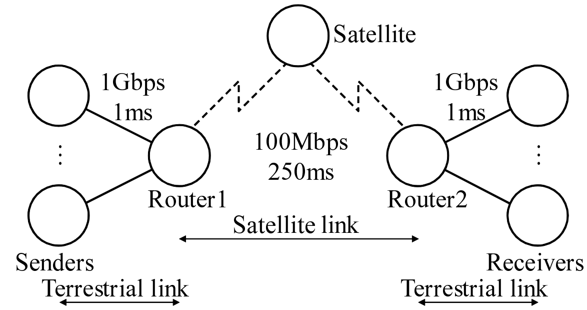

Fig. 1. Network model.

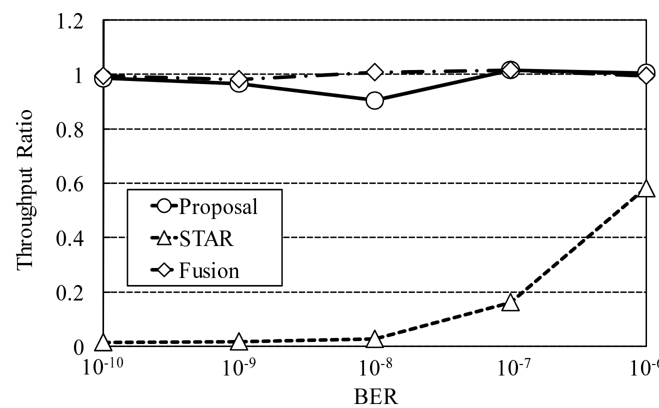

Fig. 2. Throughput ratio when BER changes and the number of total flows N is 20.

B. Simulation Results

Fig.2 and Fig.3 show the throughput ratio and the bandwidth utilization when BER (Bit Error Rate) changes, respectively. In Fig.2 and Fig.3, the number of total flows N is 20.

From Fig.2, TCP-STAR presses the throughput of coexisting TCP-NewReno flows, because TCP-STAR tries to keep the transmission rate high. On the other hand, the proposed method can obtain high friendliness same as TCP-Fusion. Since the proposed method can decrease the congestion window effectively by using the method of TCP-Fusion when TCP-NewReno coexists, the proposed method does not affect coexisting TCP-NewReno flows.

From Fig.3, when BER is larger than 10^{-8}, TCP-Fusion cannot obtain high bandwidth utilization. On the other hand, the bandwidth utilization of proposed method is larger than that of TCP-STAR when BER is from 10^{-10} to 10^{-7}. Furthermore, the bandwidth utilization of the proposed method is 1.6 times larger than that of TCP-Fusion when BER is 10^{-6}. That is, the proposed method can keep the congestion window high in case of high BER and increase the congestion window quickly in case of high propagation delay by using CWS and LWC.

Next, this paper evaluates the proposed method when the number of total flows increases. Fig.4 and Fig.5 show the throughput ratio and the bandwidth utilization when BER (Bit Error Rate) changes and the number of total flows N is 40,

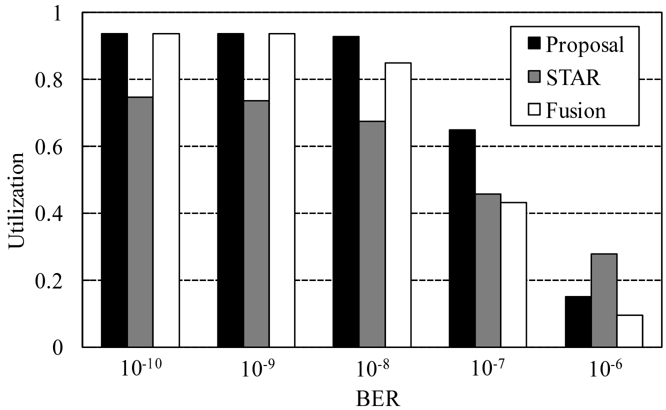

Fig. 3. Bandwidth utilization when BER changes and the number of total flows N is 20.

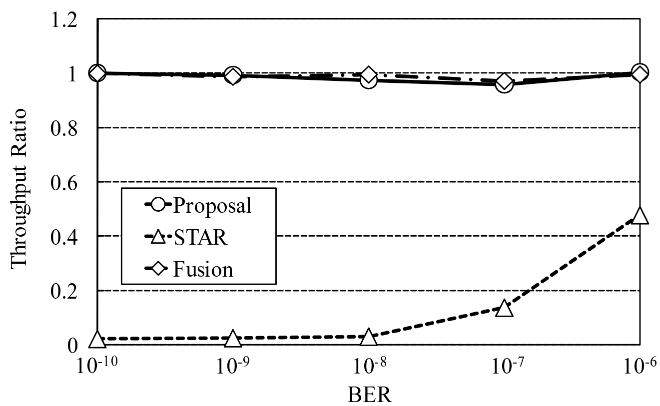

Fig. 4. Throughput ratio when BER changes and the number of total flows N is 40.

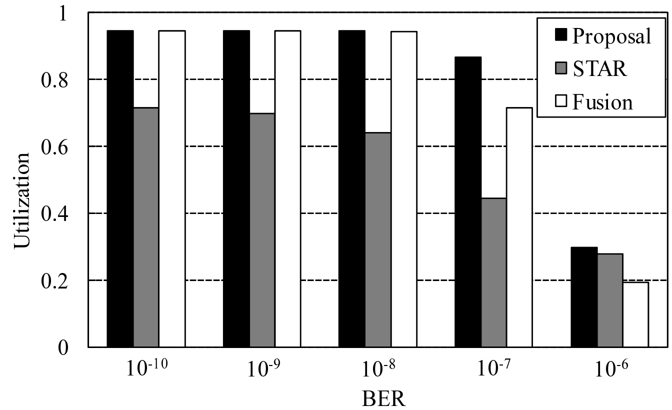

Fig. 5. Bandwidth utilization when BER changes and the number of total flows N is 40.

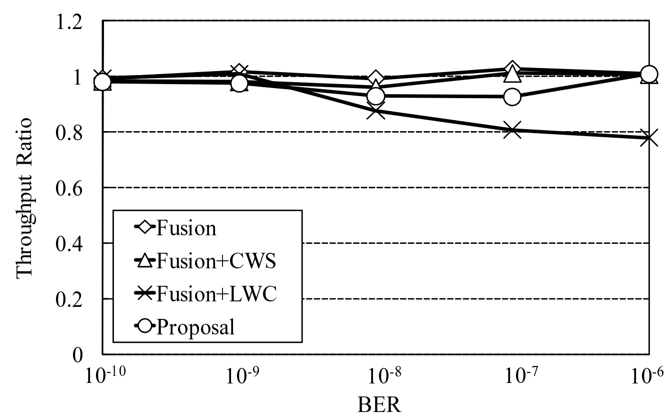

Fig. 6. Evaluation of each mechanisms in the proposed method: Throughput ratio when BER changes and the number of total flows N is 20.

respectively.

From Fig.4, the proposed method can obtain high friendliness even if the number of total flows increases. Next, from Fig.5, the bandwidth utilization of the proposed method becomes larger than that of the number of total flows is 20 when BER is 10^{-7} and 10^{-6}. Especially, when BER is 10^{-6}, the bandwidth utilization of the proposed method is larger than that of TCP-STAR. Note that the bandwidth utilization of the proposed method is smaller than that of TCP-STAR in case of $N = 20$. In this case, since the congestion becomes heavily, it is assumed that both the TCP-Fusion's control and CWS and LWC in the proposed method works well. As a result, the bandwidth utilization of the proposed method increases.

Finally, this paper evaluates the effect of each mechanisms in the proposed method. In this evaluation, we compare the proposed method with the following methods; TCP-Fusion with CWS (TCP-Fusion+CWS), TCP-Fusion with LWC (TCP-Fusion+LWC), and TCP-Fusion. Here, TCP-Fusion+CWS and TCP-Fusion+LWC are subsets of the proposed method. Fig.6 and Fig.7 show the throughput ratio and the bandwidth utilization when BER (Bit Error Rate) changes and N is 20, respectively. From Fig.6, TCP-Fusion+CWS can obtain the high friendliness like TCP-Fusion. On the other hand, the friendliness of TCP-Fusion+LWC decreases when BER is larger than 10^{-8}.

Next, from Fig.7, both TCP-Fusion+CWS and TCP-Fusion+LWC can obtain higher bandwidth utilization than TCP-Fusion when BER is larger than 10^{-8}. Since CWS sets the congestion window as large as the available bandwidth when packet losses by bit error occur, TCP-Fusion+CWS can obtain higher throughput than TCP-Fusion. As a result, the bandwidth utilization of TCP-Fusion+CWS is higher than that of TCP-Fusion. Furthermore, the bandwidth utilization of TCP-Fusion+LWC is larger than that of TCP-Fusion+CWS, because TCP-Fusion+LWC can keep the congestion window high by using LWC. However, since TCP-Fusion+LWC tends to compres the coexisting flows, the friendliness of TCP-Fusion+LWC decreases. In summary, it is found that the results of the proposed method which has CWS and LWC refrect the each effects of CWS and LWC. That is, the proposed method can obtain higher friendliness than TCP-Fusion+LWC and better bandwidth utilization than TCP-Fusion+CWS.

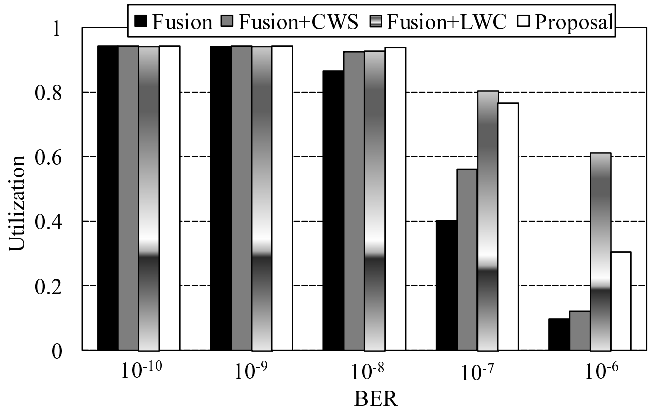

Fig. 7. Evaluation of each mechanisms in the proposed method: Bandwidth utilization when BER changes and the number of total flows N is 20.

V. Conclusion

This paper propsed a new TCP congestion control method for improving TCP friendliness over the satellite Internet. From the simulation results, we found that the proposed method can obtain high throughput and provide high friendliness over the satellite Internet.

Acknowledgments

This research was partly supported by the Ministry of Internal Affairs and Communications of Japan under Grant-in-Aid for SCOPE, the Ministry of Education, Culuture, Sports, Science and Technology, Grant-in-Aid for Young Scientists (B) 19760260, and Hiroshima City University Grant for Special Academic Research 6112.

References

[1] JAXA (Japan Aerospace Exploration Agency), “Kizuna (WINDS),” http://winds-ets8.jaxa.jp/winds/en/index.html/, 2009.

[2] Y. Nakamura, “WINDS overview,” Proc. The Internet Symposium for Application of “Kizuna” of Wideband InterNetworking engineering test and Demonstration Satellite (WINDS), pp.46-54, 2007.

[3] H. Obata and K. Ishida, “Performance Evaluation of TCP Congestion Control Method for High-speed Satellite Internet,” Expert’s Meeting for the Smooth Enforcement of the Satellite Application Experiment using WINDS, pp.71-75, 2007.

[4] M. Allman, V. Paxson, and W. Stevens, “TCP Congestion Control,” RFC2581, 1999.

[5] H. Obata, S. Takeuchi, and K. Ishida, “A New TCP Congestion Control Method Considering Adaptability over Satellite Internet,” Proc. 4th International Workshop on Assurance in Distributed Systems and Networks, pp.75-81, 2005.

[6] H. Obata, K. Ishida, S. Takeuchi, and S. Hanasaki, “TCP-STAR: TCP Congestion Control Method for Satellite Internet,” IEICE Transactions on Communications, Vol.E89-B, No.6, pp.1766-1773, 2006.

[7] H. Obata and K. Ishida, “Performance Evaluation of TCP Variants over High Speed Satellite Links,” Proc. 25th AIAA International Communications Satellite Systems Conference, no.AIAA2007-3156(14page), 2007.

[8] K. Kaneko and J. Katto, “TCP-Fusion: A Hybrid Congestion Control Algorithm for High-speed Network,” Proc. PFLDnet, pp.31-36, 2007.

[9] S. Nishimoto, H. Obata, and K. Ishida, “TCP Congestion Control Method Improving Friendliness over Satellite Links,” IEICE General Conference, BS-5-5, 2009 (in Japanese).

[10] Network Simulator - ns (version 2), http://www.isi.edu/nsnam/ns/, 2009.

[11] N. Sato, M. Kunishi, and F. Teraoka, “TCP-J: New Transport Protocol for Wireless Network Enviroments,” IPSJ Journal, Vol.43, No.12, pp.3848-3858, 2002 (in Japanese).

[12] S. Saito and F. Teraoka, “Implementation, Analysis and Evaluation of TCP-J: A New Version of TCP for Wireless Networks,” IEICE Transactions on Communications, Vol.J87-D-1, No.5, pp.508-515, 2004 (in Japanese).

[13] C. Casetti, M. Gerla, S. Mascolo, M. Y. Sanadidi, and R. Wang, “TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links,” Proc. ACM Mobicom 2001, pp.287-297, 2001.

[14] R. Wang, M. Valla, M. Y. Sanadidi, B. K. F. Ng, and M. Gerla, “Efficiency/Friendliness Tradeoffs in TCP Westwood,” Proc. IEEE SCC 2002, pp.304-311, 2002.

[15] V. Jacobson, R. Braden, and D. Borman, “TCP Extentions for High Performance,” RFC1323, 1992.

[16] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, “TCP Selective Acknowledgment Options,” RFC 2018, 1996.

[17] M. Allman, D. Glover, and L. Sanchez, “Enhancing TCP over Satellite Channels Using Standard Mechanisms,” RFC 2488, 1999.