Robust End-to-End Available Bandwidth Estimation for Interactive Satellite networks

Minsu Shin, Mankyu Park, Deockgil Oh, Byungchul Kim, and Jaeyong Lee, *Member, IACSIT*

***Abstract*—This paper presents an efficient and accurate available bandwidth estimation algorithm for satellite links which generally have long propagation delay. Either the method using the rate of returning ACKs or the method using timestamp option is adopted for the bandwidth estimation algorithms to enhance TCP performance in the literature. The method using the rate of returning ACKs, which is the most popular for the available bandwidth estimation for a TCP connection, is known to have the influences from the reverse link (TCP receiver to TCP sender) traffic variation, especially in the asymmetric satellite links. The bandwidth estimation method using packet sending time information from the TCP timestamp option also has the drawback that the influence of the bottleneck link presented in the forward path is not properly considered. To overcome these limits, our algorithm performs in two steps. A forward path delay is calculated at the sender with the receiver sending time for an ACK packet in the first step. And in the second step, the forward path delay is updated more delicately through removing unrelated part from the forward path delay, and is applied for available bandwidth estimation. The simulation results show that our algorithm can avoid the influence of the traffic variation in the return path (e.g., ACK compression, DAMA delay), and estimate the available bandwidth over the forward path more accurately.**

***Index Terms*—Bandwidth Estimation, Satellite TCP, Wireless TCP.**

I. Introduction

Satellite networks are becoming the important candidate in information and communication infrastructures because they provide several advantages to the wireless Internet communications. They generally provide wider coverage, higher bandwidth, and relatively faster establishment than other terrestrial wireless networks. Nevertheless, satellite links pose some challenges to the congestion control operation of the TCP (Transmission Control Protocol) [10]. Main challenges of the satellite networks to the TCP performance are the long RTT (round-trip time) which is generally around 600msec in the bi-directional networks over the geostationary satellite, and the presence of the high packet loss rate by wireless random error which causes congestion control of the TCP spuriously.

To solve the problems while keeping end-to-end semantics of TCP, an accurate bandwidth estimation algorithm is required to control congestion window (cwnd) value correctly according to the changes of the network situation. A number of the contributions which estimate the available bandwidth on the links and apply it to the congestion control algorithm of a TCP connection have been proposed so far. The available bandwidth estimation algorithms in the literature could be classified into two categories, i.e., ACKs inter-arrival time based scheme and Timestamp option based scheme. Fig. 1 shows the estimation performance of the TCP Westwood [2] which adopts ACKs inter-arrival time based scheme and the TCP New Jersey [4] adopting timestamp option based scheme as well as TCP NewReno and TCP Vegas. It is the simulation result for the goodput performance when there is a background traffic varying 0-5Mbps on the reverse path of the bottleneck link of 5Mbps. From the comparison, TCP NewReno does not have significant impact from the reverse cross traffic because it is not using bandwidth estimation for its congestion control, even though it has the lowest goodput performance regardless of the amount of the reverse cross traffic. While TCP Vegas has the best performance when the traffic load is light on the link, it suffers drastic performance degradation from the high traffic load because of the nature of the delay based congestion control algorithm. TCP New Jersey provides more stable result than TCP Westwood when there is cross traffic on the reverse path. However, TCP New Jersey is still missing some aspects which have to be additionally considered for the accurate estimation. This paper identifies these values which are involved for the available bandwidth estimation process unnecessarily, and presents the enhanced end-to-end available bandwidth estimation algorithm giving better performance by excluding these unrelated values.

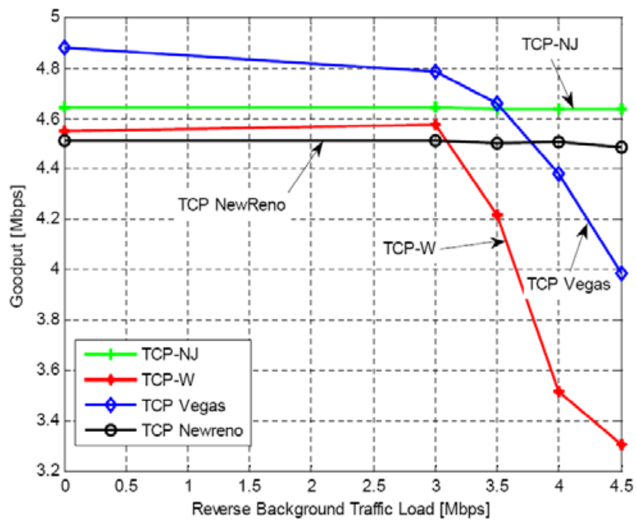


Fig. 1. Impact of the Reverse background traffic on Goodput Performance of TCP variants

Manuscript received August 31, 2011; revised September 14, 2011.

M. Shin, M. Park and D. Oh are with the Electronics and Telecommunications Research Institute (ETRI), Daejeon, 305-700 KOREA (e-mail: msshin@etri.re.kr, neomkpark@ etri.re.kr, dgoh@ etri.re.kr).

B. Kim and J. Lee are with Dept. Information Communication Engineering, Chungnam National University, Daejeon, 305-764 KOREA (e-mail: byckim@cnu.ac.kr, jyl@cnu.ac.kr).

II. RELATED WORKS

In the paper, we are interested in the passive scheme that is not using special probe packets for available bandwidth estimation. And among many algorithms in this category, we observe the ACKs inter-arrival time based schemes [2,3] and Timestamp option based scheme [4].

A. TCP Vegas

TCP Vegas [11] proposes a new sender-based congestion control mechanism aiming to prevent congestion events by monitoring the difference between the expected rate and the actual rate. In fact, TCP Vegas congestion control is quite different from TCP Reno, TCP Reno uses the loss of packets as a signal that there is congestion in the network, so TCP Reno needs to create losses to find the available bandwidth of the connection. In contrast, the goal of TCP Vegas is to pro-actively detect congestion in its incipient stages, and then reduce sending rate in an attempt to prevent the occurrence of packet losses.

To detect network congestion, once every round trip time (*RTT*), TCP Vegas uses the current window size (*W*), the most recent RTT (*RTT*), and the minimum RTT observed so fat (*baseRTT*) to compute:

$$\begin{aligned} diff &= \left(\frac{W}{baseRTT} - \frac{W}{RTT}\right) \cdot baseRTT \\ &= W \cdot \frac{RTT - baseRTT}{RTT}. \end{aligned} \tag{1}$$

Since (*RTT-baseRTT*) is the total path queuing delay and W/RTT is an estimate of the current throughput, the product of these two values is an estimate of the number of packets from this flow that are backlogged in the network. The goal of the Vegas congestion avoidance algorithm is to keep this number within a fixed range defined by two thresholds, α and β. Thus, once every RTT when not in Slow Start phase, TCP Vegas adjusts the window size as follows:

$$W = \begin{cases} W+1 & , diff < \alpha \\ W & , \alpha < diff < \beta. \\ W-1 & , diff > \beta \end{cases} \tag{2}$$

Another feature of TCP Vegas is its modified Slow Start behavior, which is more conservative than TCP Reno's. Specifically, If the standard Slow Start mechanism is not limited by buffer dimensions at the end hosts, the congestion window size will double every RTT (if there are no losses), until an overrun of connection capacity. Therefore, the losses may be on the order of half the current congestion window. TCP Vegas prevents congestion events during the Slow Start allowing an exponential window growth only every other RTT. During this time, the congestion window is fixed. Thus, a valid comparison of the expected and actual rates can be performed.

B. TCP Westwood

TCP Westwood [2] aims at limiting the consequences of the losses introduced by a wireless channel, which are always erroneously ascribed to congestion by the standard TCP protocol. To this end, instead of halving the congestion window after three duplicate ACKs, and fixing the slow start threshold to this value, TCP Westwood sets ssthresh adaptively according to the Bandwidth Estimator (BE), avoiding the dramatic slow-down of the transmission rate of the TCP standard versions. The bandwidth is estimated by averaging the rate of returning ACKs. In particular, the reception of an ACK at the time t_k implies that an amount of data d_k has been received. Therefore, the *k*-th sample of bandwidth used by a given connection is measured as

$$b_k = \frac{d_k}{t_k - t_{k-1}}. \tag{3}$$

These values are then filtered to obtain the available bandwidth estimate (BWE) at time t_k, given by

$$\hat{b}_k = \alpha_k \cdot \hat{b}_{k-1} + (1-\alpha_k)\left(\frac{b_k + b_{k-1}}{2}\right). \tag{4}$$

After loss detection, TCP Westwood sets the ssthresh and cwnd as follows

$$\begin{aligned} ssthresh &= \hat{b} \times RTT_{\min} \\ cwnd &= \begin{cases} ssthresh & \text{if cwnd} > \text{ssthresh} \\ 1 & \text{after a timeout} \end{cases} \end{aligned} \tag{5}$$

Several variants of the basic algorithm have been presented in the literature to improve performance by modify its bandwidth estimation method and start-up procedure.

C. TCP Jersey and TCP New Jersey

TCP Jersey [3] was proposed as a new TCP scheme which is capable of distinguishing the wireless packet losses from the congestion packet losses. TCP Jersey consists of two components for loss differentiation; the available bandwidth estimation (ABE) and the congestion warning (CA) router configuration. Basically, TCP Jersey also uses the rate of the returning ACKs for the ABE as in the TCP Westwood. On the receiving 3 DupACKs, TCP Jersey decides whether there is congestion on the network according to the congestion warning information sent from the router. It controls its sending rate depending on the estimate of available bandwidth only when the loss was caused by the congestion. In TCP Jersey, TimeSliding Window(TSW) Estimator is adopted for the ABE.

$$R_n = \frac{RTT \times R_{n-1} + L_n}{(T_n - T_{n-1}) + RTT}. \tag{6}$$

The optimum congestion window (ownd) in units of segment is then calculated as in (7).

$$ownd = \frac{RTT \times R_n}{seg\_size}. \tag{7}$$

TCP Jersey estimates the available bandwidth in this way once every RTT period. TCP New Jersey [4] adopted Timestamp-based available bandwidth estimator (TASE) instead of ABE to enhance the performance of the ABE algorithm in TCP Jersey. TCP New Jersey takes the same equation for bandwidth estimation and *ownd* calculation as in (4) and (5) except that t_n becomes the arrival time of the *n*-th packet at the receiver. The timestamps, which are delivered by ACKs and received at the sender side, closely reveal the arrival traffic pattern at the receiver side. The estimated available bandwidth by TABE is thus less affected by the reverse link conditions and immune to the ACK loss on the reverse link.

III. Proposed Enhancement

It is needed to consider only variations on the forward path to control accurately the sending rate of the TCP. However, the ACK based schemes such as packet pair and packet trains [6], are affected by the network congestion because they use the rate of returning ACKs. This causes the inaccurate estimation of the available bandwidth and finally fails to have the expected enhancement of TCP performance. To limit the effect of the network status on the reverse path, timestamp based scheme which utilizes packet inter-arrival time at the receiver has been proposed [4]. Through the simulation, it is shown that the timestamp based scheme provides more accurate and more stable estimation performance in the case of the presence of cross traffic over the reverse path. However, even though this type of estimation scheme has its own advantages, there still presents the unrelated components to the packet transmission dispersion over the path within the packet inter-arrival time itself. Fig. 2 illustrates sending and receiving time of the data packets and ACKs between TCP sender and receiver.

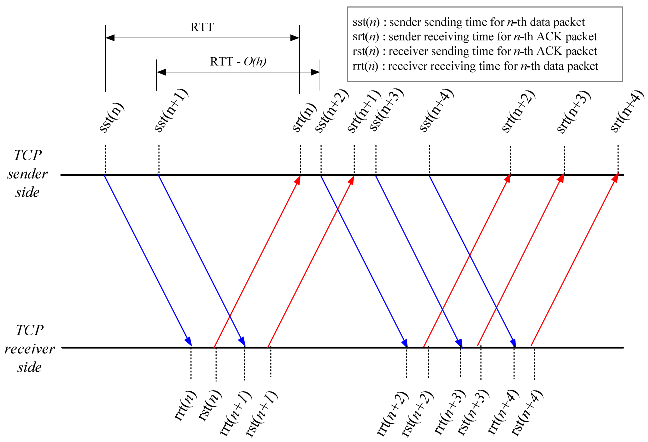


Fig. 2. Time diagram example of TCP flow

When we consider basic packet pair concept, packet inter-arrival time of a certain packet pair measured at the receiver can be represented as in (8).

$$\Delta T_{rrt}(n) = \Delta T_{sst}(n) + \Delta t_1 + \Delta t_2 . \tag{8}$$

where $\Delta T_{rrt}(n)$ is packet inter-arrival time at the receiver, i.e., *rrt(n)-rrt(n-1)*, *rrt(n)* is the receiver receiving time of *n*-th data packet, $\Delta T_{sst}(n)$ is packet inter-sending time at the sender, i.e., *sst(n)-sst(n-1)*, *sst(n)* is the sender sending time of *n*-th data packet, Δt_1 is the variation expected by the narrow link over the forward path and therefore should be measured accurately for the available bandwidth estimation, and Δt_2 is the scheduling delay at the transmitter of the TCP sender which could be considered to be negligibly small value.

The available bandwidth over the forward path can be calculated by using TCP receiver's timestamp value received at the sender, as in (9).

$$R_n = \frac{RTT \times R_{n-1} + L_n}{(rrt_n - rrt_{n-1}) + RTT} . \tag{9}$$

However, it should be noticed that the packet inter-sending time at the TCP sender, *sst(n)-sst(n-1)*, in (6) includes the variation caused by the operational condition within the sender rather than any transmission delay. These timing variations listed below have to be excluded in the available bandwidth estimation.

- *In the case of estimation for the first packet in a new RTT period* (Δt_3) : this can happen when the estimation is carried out using the first packet in a new RTT period and the last packet in the previous RTT period. This is the case of *{sst(n + 2) - sst(n + 1)}* in Fig. 2. This value could be large, especially in the interactive satellite network because of its long RTT which is typically more than 500 ms. Moreover if the possible delay for MAC resources for return-link transmission is considered, this value becomes larger.
- *In the case for the TCP sender to wait for data to send* (Δt_4) : this can happen when there is no data to send at the TCP layer even if cwnd is enough large. This is not the case that the packet dispersion is varied because of a network condition changes.

Although the second case, Δt_4, is often ignored in the experiments assuming that TCP sender always has data to send because it is hard to estimate. In the paper, we consider these uncorrelated parts to the transmission delay in our available bandwidth estimation process. We call these parts of the delay *the waiting time*.

From this observation, we present, in the paper, the enhanced available bandwidth estimation scheme that can provides more accurate result by removing this waiting time from the packet inter-sending time at the TCP sender. Based on the idea that the waiting time, i.e., Δt_3 and Δt_4, could have a large value, but it would have low occurrences, e.g., once every RTT, it can be calculated as in (10).

$$\begin{aligned} \Delta T(n) &= \left| \Delta T_{sst}(n) - \Delta \overline{T}_{sst}(n) \right| \\ \Delta \overline{T}_{sst}(n) &= \alpha \times \Delta T_{sst}(n) + (1-\alpha) \times \Delta \overline{T}_{sst}(n-1) \end{aligned} \tag{10}$$

where $\Delta T_{sst}(n)$ indicates packet inter-sending time *{sst(n) - sst(n-1)}*, $\Delta \overline{T}_{sst}(n)$ is the average of the packet inter-sending times upon receiving *n*-th packet at the receiver. EWMA (Exponentially Weighted Moving Average) is applied to obtain the average, $\Delta \overline{T}_{sst}(n)$. The available bandwidth can be estimated by modifying (9) with adopting (10), like in (11).

$$R_n = \frac{RTT \times R_{n-1} + L_n}{(rrt_n - rrt_{n-1}) + \Delta T(n) + RTT} . \tag{11}$$

And the optimum congestion window (ownd) in our scheme is calculated in the same way as in (7).

IV. Performance Evaluation

In this section, we evaluate the performance of the proposed algorithm under various conditions. We have conducted the comparison of our scheme with TCP Westwood and TCP New Jersey to evaluate the estimation performance in the several scenarios. All the simulation is performed using NS-2 simulator [7].

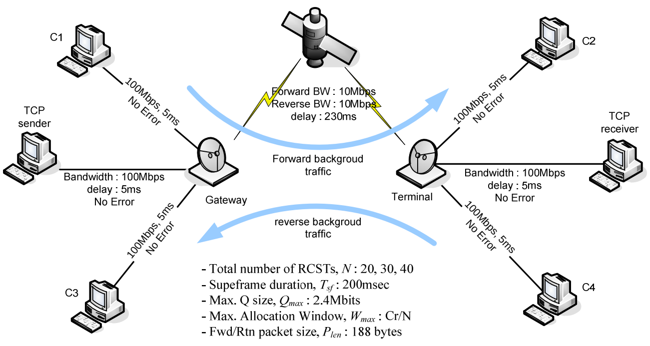


Fig. 3. Simulation topology

TABLE I: PARAMETERS USED IN THE SIMULATIONS

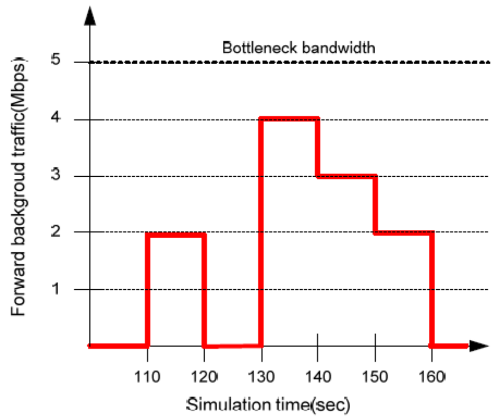
Parameters	Value
Return link Capacity, C_r	10 Mbps
# of RCSTs, N	20, 30, 40
Total Propagation Delay, τ	480 ms
Superframe duration, T_{sf}	200 ms
Max. Queue Size, Q_{max}	2.4 Mbits
Max. Allocation Window, W_{max}	C_r/N
Forward/Return link packet size, P_{len}	188 bytes

Fig. 3 depicts the simulation topology used in the performance evaluation. The delay in the simulation means one-way propagation delay of the corresponding link and the buffer size at each link takes the value of the bandwidth-delay product (BDP) of the link. Table I describes simulation parameters used for the performance evaluations which generate more realistic scenarios.

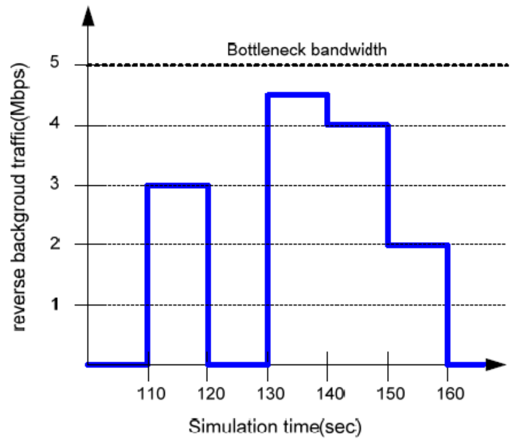
Our scheme focuses on limiting an impact of the reverse link variation to the available bandwidth estimation performance and excluding so called *the waiting time* inevitably included in the transmission delay. To recognize this property, we evaluate the estimation performance in two different environments where there is CBR background traffics (a) only in the forward direction and (b) both in the forward and reverse directions. The background traffic patterns used in the evaluations for each direction are as in Fig. 4.

A. *Impact of Forward Background Traffic*

First, we want to evaluate the performance of our estimator when there is only forward background traffic with the amount in Fig. 4(a). The goal of the evaluation is to see the performance of the proposed algorithm itself without any other element affecting the performance since the algorithm can estimate the pure forward link variations. We compare the estimation performances of the BE scheme used in the TCP-W, TSW scheme of the TCP-NJ and our scheme called Enhanced TSW (E-TSW). As shown in the Fig. 5, BE tends to follow the changes of the available bandwidth slowly and shows variations at where the available bandwidth changes abruptly. Moreover, it seems to have some fluctuations even in the steady period (130-140sec period in Fig. 5). On the contrary, TSW scheme performs well across all rage of the simulation and responds very quickly to the network state changes. However it still has some fluctuations especially in congested situation. E-TSW shows the best performance in this scenario and it responds to the change of the available bandwidth quickly and accurately.



(a) Forward Background Traffic Load(Mbps)



(b) Reverse Background Traffic Load(Mbps)

Fig. 4. Background traffic pattern

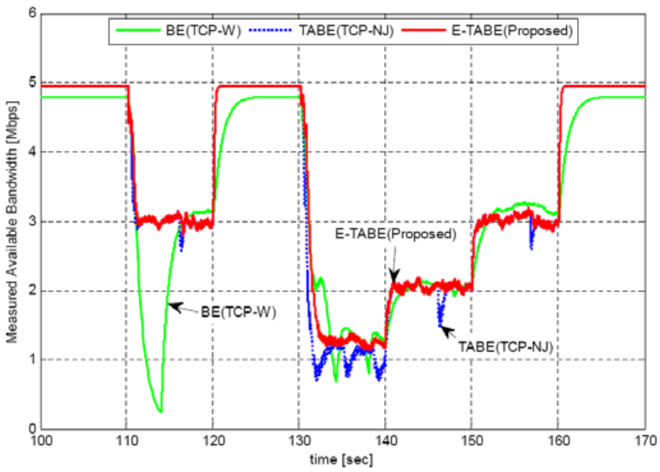


Fig. 5. ABE performance with background traffic only in the forward link

B. *Impact of Forward Background Traffic*

Here, we present the performance of the estimators under consideration when there are cross traffics both in the forward and reverse directions in the same way as in Fig. 4. As shown in Fig. 6, BE scheme shows much more unstable performance when there is a reverse cross traffic as compared with the case where there is only forward background traffic.

This is the expected results because the BE scheme uses returning ACK rate for estimation, which can be naturally affected by the reverse link conditions. Since TSW and E-TSW schemes use the packet arrival time at the receiver for the estimation, their performances are basically immune to the variations in the reverse link. However, from the simulation result we see that the E-TSW scheme provides more stable performance than the TSW scheme because it removes unwanted elements that could affects the estimation performance.

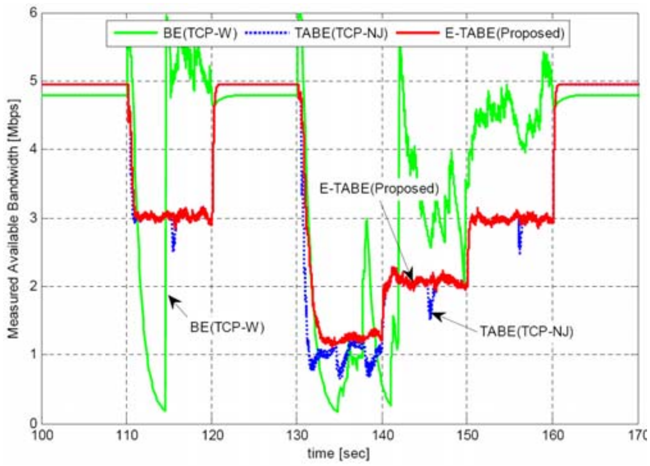


Fig. 6. ABE performance with background traffic both in the forward and the reverse link

V. Conclusion

In this paper, we evaluated the previous available bandwidth estimation schemes under the network conditions where there is the background traffics on the reverse link and showed that some previous schemes have the performance degradation in terms of throughput in those environments. Especially for the scheme using the rate of returning ACKs such as TCP-W, the performance is significantly affected by the traffic conditions over the reverse link because the inter-arrival times of the successive ACKs could be impacted by the background traffics on the reverse link even though they have to only reflect the variation of the available bandwidth on the forward link. TCP-NJ takes the packet inter-arrival time at the receiver as a measure for bandwidth estimation so that it can be immune to the fluctuation of the reverse traffic. However, we found that the TSW scheme used in TCP-NJ still has the elements to be removed in the estimation process because they are not related to the packet delivery time over a link.

We proposed a new scheme for the available bandwidth estimation to solve the problem, thereby improving the estimation performance in more realistic environment. Through the simulations with different scenarios, we verified that our scheme has better performance in the conditions with severe background traffic load over the reverse link in terms of responsiveness and stability.

Acknowledgment

This work was supported by the IT R&D program of KCC[2009-S-039-02, Development of Satellite Return Link Access Core Technology for High Efficient Transmission].

References

[1] C. Caini, R. Firrincieli, and D. Lacamera, “Comparative Performance Evaluation of TCP variants on Satellite Environments”, in *Proc. IEEE Int. Conf. on Communications.(ICC)*, Dresden, Germany, pp. 1-5, June 2009.

[2] C. Casetti, M. Gerla, S. Mascolo, M. Y. Sanadidi, and R. Wang, “TCP Westwood: end-to-end congestion control for wired/wireless networks”, in *Wireless Networks*, Kluwer Academic Publisher, vol.8, pp. 467-479, 2002.

[3] K. Xu, Y. Tian, and N. Ansari, “TCP-Jersey for wireless IP communications”, *IEEE J. Sel. Areas Commun.*, vol. 22, pp. 747-756, 2004.

[4] K. Xu, Y. Tian, and N. Ansari, “Improving TCP performance in integrated wireless communications networks”, *Computer Networks*, vol. 47, pp. 219-237, 2005.

[5] M.Allman et al., “Ongoing TCP Research Related to Satellites”, IETF RFC2760, Feburary 2000.

[6] R. S. Prasad, M. Murray, C. Dovrolis, K. Claffy, “Bandwidth Estimation: Metrics, Measurement Techniques, and Tools”, *IEEE Network*, November/December 2003.

[7] The Network Simulator – NS-2, http://www.isi.edu/nsnam/ns/

[8] V. Jacobson, R. Braden, D. Borman, ”TCP extensions for high performance,” IETF RFC 1323, May 1992.

[9] Vern Paxson, “End-to-End Internet Packet Dynamics,” *IEEE/ACM Trans. Networking*, vol. 7, no. 3, pp. 277-292, June 1999.

[10] Y. Hu and V. O. H. Li, "Satellite-based internet : a tutorial," *IEEE Commun. Mag.*, vol. 39, no. 3, pp. 164-171, Mar. 2001.

[11] L. Brakmo and L. Peterson, “TCP Vegas: end to end congestion avoidance on a global Internet,” *IEEE J. Sel. Areas Commun.*, vol. 13, no. 8, October 1995, pp. 1465~1480.

[12] E. Goldoni and M. Schivi, “End-to-End Available Bandwidth Estimation Tools, an Experimental Comparison,” in *Proc. 2nd COST Traffic Monitoring and Analysis(TMA) Workshop*, April 2010.

[13] C. D. Guerrero, M. A. Labrador, “On the applicability of available bandwidth estimation techniques and tools,” *Computer Communications*, vol. 33, no. 1, pp. 11–22, 2010.

[14] N. Hu and P. Steenkiste, “Evaluation and Characterization of Available Bandwidth Probing Techniques,” *IEEE J. Sel. Areas Commun.*, vol. 21, no. 6, pp. 879–894, 2003.

[15] L. Angrisani, S. D’Antonio, E. Esposito, and M. Vardusi, “Techniques for available bandwidth measurement in IP networks : a performance comparison,” *Computer Networks*, vol. 50, no. 3, pp. 332–349, 2006.

[16] N. Parvez and E. Hossain, “TCP Prairie : a sender-only TCP modification based on adaptive bandwidth estimation in wired-wireless networks,” *Computer Communications*, vol. 28, no. 2, pp. 246-256, 2005.

[17] J. Navratil and R. L. Cottrell, “ABwE: A Practical Approach to Available Bandwidth,” in *Proc. 4th International workshop on Passive and Active network Measurements (PAM2003)*, California, USA, 2003.

[18] J. Choi and C. Yoo, “One-way delay estimation and its application,” *Computer Communications*, vol. 28, no. 7, pp. 819-828, 2005

[19] A. Capone, L. Fratta, and F. Martignon, “Bandwidth Estimation Schemes for TCP over Wireless Networks,” *IEEE Trans. Mobile Computing*, vol. 3, no, 2, pp. 129-143, 2004.

[20] J. Liebeherr, M. Fidler, and S. Valaee, “A System-theoretic approach to bandwidth estimation,” *IEEE/ACM Trans. Networking*, vol. 18, no. 4, pp. 1040-1053, 2010.

Minsu Shin received the B.S. and M.S. degrees in electronics engineering from Korea Aerospace Univ., Seoul, Korea in 1998 and 2000, respectively, and the Ph.D degree in information and communication engineering from Chungnam National Univ., Daejeon, Korea in 2011. Since he joined the Electronics and Telecommunications Research Institute (ETRI) in 2000, he has worked for satellite communication systems and broadcasting systems until now. Now his research interests are satellite communication network design, wireless

resource management, network planning, wireless TCP and cross-layer enhancement.

Mankyu Park received the B.S. and M.S. degrees from the Department of Information and Communication Engineering, Kongju National University, Korea in 1999 and 2001, respectively, and the Ph.D degree in information and communication engineering from Chungnam national university in 2011. He worked as a system programmer at venture companies named Syswill and KDnet from 2001 to 2005. Now he is a senior researcher in Satellite Broadcasting & Telecommunications Convergence Research Team at ETRI. His current research activities include MAC for interactive satellite network, congestion control and QoS issues.

Byungchul Kim received the B.S. degree in electronics engineering from Seoul National University and M.S. and Ph.D degrees in electronic engineering from Korea Advanced Institute of Science and Technology (KAIST), Korea, in 1988, 1990 and 1996. He is currently a professor at the Department of Information and Communication Engineering of Chungnam National University, Korea since 1999. Also, from 1993 to 1999, he worked as a Research Engineer at the Samsung Electronics. His research interests include computer networks, wireless internet, sensor networks, mobile communications and wireless mesh networks.

Jaeyong Lee received the B.S. degree in electronics engineering from Seoul National University and M.S. and Ph.D degrees in electronic engineering from Korea Advanced Institute of Science and Technology (KAIST), Korea, in 1988, 1990 and 1995. He is currently a professor at the Department of Information and Communication Engineering of Chungnam National University, Korea since 1995. Also, from 1990 to 1995, he worked as a research engineer at the Digicom Institute of Information and Communications. His research interests include Internet protocols, traffic control, performance analysis, wireless mesh networks and mobile internet.

Deockgil Oh received the B.S., M.S. and Ph.D degrees in electronics engineering from Seoul National University (SNU) in 1980, 1984 and 1996, respectively. He joined Electronics and Telecommunications Research Institute (ETRI) in 1982, where he is currently working as a director in Dept. of Satellite Broadcasting and Communication Convergence. His research interests include wireless access technology, mobile communication and broadcasting system and future generation satellite broadcasting architectures.