TCP Congestion Window Tuning for Satellite Communication Using Cross-Layer Approach

Jong-Mu Kim
Department of Electrical and Computer Engineering
Ajou University
Suwon, Korea
blackkim822@ajou.ac.kr

Seung-Yong Lee
Department of Netwokr Centric Warfare
Ajou University
Suwon, Korea
winstar@naver.com

Jae-Hyun Kim
Department of Electrical and Computer Engineering
Ajou University
Suwon, Korea
jkim@ajou.ac.kr

***Abstract*— In this paper, we propose a TCP congestion window (CWND) tuning scheme which considers the resource allocation information of satellite link and the number of sessions. The proposed scheme sets TCP congestion window size by using satellite resource allocation information exchanged between TCP and the link layer and the number of session. To evaluate the performance, we implemented the proposed scheme on TCP Linux kernel over TCP-splitting based performance enhancing proxy (PEP) test-bed. The results show that the performance of proposed protocol better than existing TCP protocols, both in single and multiple sessions in variant bit error rates (BER) notably in higher BER.**

***Keywords*— Performance Enhancing Proxy, Cross-layer, Number of Session Congestion Window.**

I. Introduction

Satellite communication system can provide world-wide communication services using wide coverage of satellites [1]. However, TCP shows low performance in satellite links because of three reasons below [2]. First, many people share limited bandwidth, so some people cannot get enough bandwidth. Second, satellite links have long round-trip time (RTT), so a long time is required to increase sufficient TCP congestion window (CWND). Last, satellites have higher bit error rates (BER) than terrestrial links, which causes unnecessary congestion control.

There are many researches on the performance enhancing proxy (PEP) to enhance the TCP performance in satellite communication [3-5]. Most PEPs use TCP splitting to isolate the satellite network from the terrestrial networks in order to exploit the advantages of advanced or customized TCP versions over satellite links. However, TCP splitting PEP still has a low performance in high BER because this approach still estimates bandwidth as in conventional TCPs.

To solve this problem, we propose a congestion window tuning scheme using a cross-layer approach between the TCP and link layer. For the tuning, the TCP uses the allocated bandwidth of satellite links and the number of sessions. The performance of this proposed scheme is evaluated using the implementation of testbed.

The remainder of this paper is organized as follows: Section II a detailed description of the proposed protocol is presented. Section III explains test-bed architecture of the proposed protocol and describes the performance evaluation of the proposed protocol. Finally, we have the conclusion.

II. Proposed Protocol

A. System Model

The protocol stack defined in this paper is shown in Fig. 1. It adapts a pair of TCP splitting connections to divide an end-to-end (E2E) TCP connection into three partitions. Terrestrial segments comply with the standard TCP protocol to ensure compatibility with the server (receiver) and the client (sender), while the TCP for the satellite segment can be customized for the satellite links. A cross-layer approach allows direct information exchange between non-adjacent layers, here between the TCP and the link layer. The link layer is logically situated directly above the physical medium, and it delivers data across the satellite link. To set the optimum TCP CWND for a satellite link, TCP requests the required information to the link layer, and link layer provides immediate feedback to TCP. In the TCP initiation procedure, the gateway and satellite terminal generate TCP splitting. For the splitting, if the PEP in the gateway gets the TCP SYN between the client and the server, the PEP sends an ACK to the server instead of the client. The gateway also opens a TCP connection, and forwards TCP SYN to the client. If this TCP SYN is received on the PEP in the satellite terminal, this PEP also sends an ACK to gateway instead of client, opens a TCP connection, and forwards the TCP SYN to the client. The impact of long RTT on the terrestrial networks is avoided using this configuration because the congestion windows for two terrestrial links and a satellite link are controlled separately. These are the advantages of implementing TCP splitting based PEP solution at the satellite legs which totally isolate the satellite channel impairments.

Frequency-time resources may be configured freely according to the standard satellite digital video broadcasting-return channel via satellite (DVB-RCS). The network control center (NCC) via demand assigned multiple access (DAMA) method shares the frequency resource-terminal time in the return link. If the satellite terminal requests resources from NCC, NCC assigns the resources and informs this in the forward link. The resource allocation is determined by the DAMA controller on the basis of the channel conditions and the available resources. Each satellite terminal has a DAMA

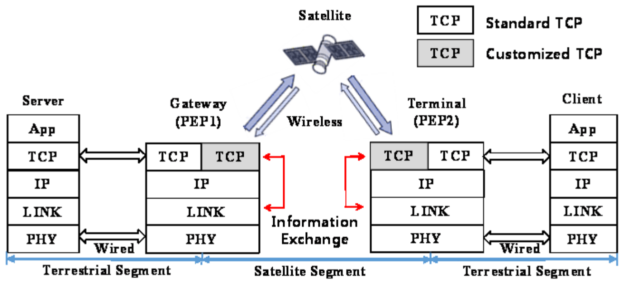

Figure 1. System model of the proposed protocol

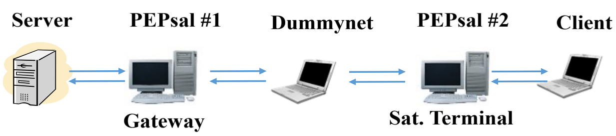

Figure 2. PEP testbed Architecture

agent at the link layer. Using the cross-layer design, TCP requests information on its demand resource allocation, called demand rate, to the link layer. The DAMA agent at the link layer assigns the request demand and allocates the resource according to it.

B. Proposed CWND Mechanism

CWND size is a key parameter that reflects the effect of network conditions. CWND adjusts the transmission rates of the TCP segment and determines quantity, and the rates of packets arriving on the link layer. However, the slow start and fast recovery for controlling CWND as in conventional TCP cannot optimize the performance of TCP because the bandwidth of satellites is not changed until the new allocation. Thus, by incorporating the demanded rate into the link layer, to adjust CWND parameter this process is capable of allocating its resources in a more intelligent fashion. Therefore, transmission rates will not unnecessarily be decreased due to packet loss on the radio link because there is less probability of packet drop by congestion on the satellite segment.

In the customized TCP of the satellite terminal, it requests a size of bandwidth from the DAMA agent at the link layer for the tuning of CWND upon reception of TCP SYN from server. Upon receiving the request from TCP, the DAMA agent sends the bandwidth allocation request message and protocol overheads to the DAMA controller of the NCC. The DAMA controller then decides the size of bandwidth and its information is reported through terminal burst time plan (TBTP) in the forward link. Upon receiving TBTP, the DAMA agent of the satellite terminal reports the size of the bandwidth to the TCP. Finally, TCP adjusts its CWND using (1),

$$CWND = \frac{RTT * R_A}{(1+K) * TCPsgmsize * N}. \qquad (1)$$

Where RTT and R_A are round-trip time and resource allocation, respectively. K is ratio of overhead including headers of the TCP/IP and the link layer, N is the number of sessions, and *TCPsgmsize* is the size of the TCP segment.

TABLE I. SETUP CONFIGURATION PARAMETER

Parameter	Value
Bandwidth	8Mbps
RTT	500ms
Maximum number of segment	326
Queue size	500 bytes

III. PERFORMANCE EVALUATION

The test-bed architecture which is implemented to conduct our experiment is shown in Fig. 2. We install PEPsal [6] on these boxes in order to act as TCP splitting to divide E2E TCP connections into three segments. PEPsal is an open source software for Linux OS. This allows for the customizing of the TCP on the satellite segment. The middle box (third from left) emulates the wireless satellite link using Dummynet [7]. Dummynet enforces queue and bandwidth limitations, delays, packet losses and multipath effects. It can run in different operating systems, but in our case it runs on FreeBSD OS. To measure network performance in our test-bed, we install iperf [8] on both the end nodes. It is a modern alternative tool written in C for network performance measurement. It measures the throughput of a network that is carrying them. The parameters configured in this experiment are as shown in Table I.

This section presents the results obtained from the conducted experiment in different test scenarios. We compared the performance of the proposed scheme with TCP variants (TCP Reno, CUBIC, Hybla) without PEP, and TCP-splitting based PEP connection with TCP Hybla in terms of the CWND and TCP throughput. The TCP variants were configured on sender and receiver for E2E TCP performance evaluation. Moreover, we evaluated TCP performance in the presence of multiple connections.

In TCP splitting, PEPsal was implemented on the gateway and satellite terminal. As connection splitting was transparent to the end users, we used TCP Reno on the terrestrial network (receiver to gateway or sender to satellite terminal). On the split connection, we enabled TCP Hybla as customized TCP. To apply the proposed scheme, we installed the proposed TCP on the satellite segment to use as customized TCP on the satellite link (replacing TCP Hybla). With this setting, we gain benefits of TCP/splitting by using the tuned CWND TCP on the radio link and improve performance.

We evaluated the proposed protocol in terms of TCP throughput as a function of the link error rate. As in Fig. 3, the proposed protocol outperforms the TCP variants and TCP-split PEP. On the satellite segment, most of the packet loss was due to the radio link error. However, standard TCP protocols have no mechanism to identify the cause of packet loss. Their congestion control mechanism was affected by radio link errors. By contrast, the use of PEP provides a benefit to adopt the customized TCP on the satellite portion which improves the

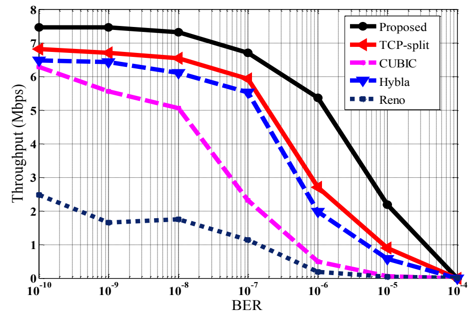

Figure 3. Average TCP throughput Vs. BER in satellite link

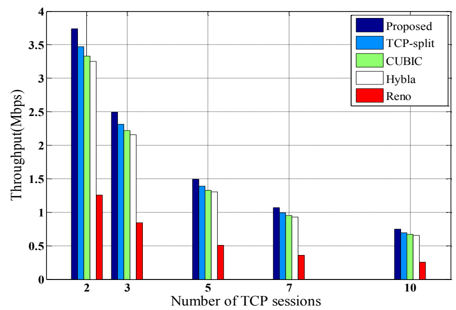

Figure 4. Average TCP throughput of multi-sessions in satellite networks (at BER = 0)

performance shown in the figure (TCP-split). In addition to TCP-split, the CWND of customized TCP was properly tuned using the size of the bandwidth allocated on the satellite link with the proposed scheme. As the link error rate increased, the throughput gained by the proposed protocol was increased.

The performance of TCP in multiple sessions (2, 3, 5, 7 and 10 connections) at error-free channel is depicted in Fig. 4. As the figure shows the proposed scheme achieved higher throughput. Similarly, we conducted an experiment with the same configuration and number TCP sessions except channel error was set to $BER = 10^{-6}$. As the Fig. 5 illustrates, the customized TCP tolerates the link error even in multiple connections. Throughput of TCP Reno and CUBIC was severely affected by the link error in all groups of the TCP sessions.

IV. Conclusion

The purpose of this paper is to improve the performance of TCP in DVB-RCS networks. As it discussed in this paper, the main reasons for the poor response of the TCP throughput over satellite links are long RTT and packet loss on the radio link. Long propagation delay directly affects E2E performance because of TCP's ACK algorithm. Link failure due to errors caused the transmission rate to decrease. Many solutions have been proposed to alleviate TCP problems in satellite communication. However, these solutions cannot mitigate all the reasons for poor TCP performance in satellite networks.

In this paper, we proposed PEP which considers satellite resource allocation information and the number of sessions. TCP tunes the CWND accurately using tuning algorithm on

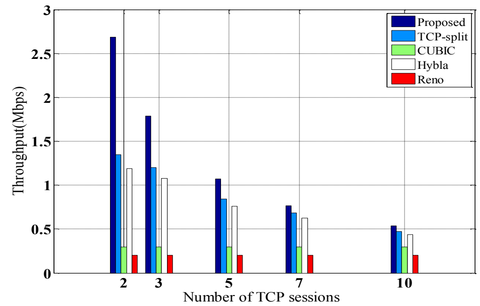

Figure 5. Average TCP throughput of multi-sessions in satellite networks (at BER = 10^{-6})

satellite segment based on the information of resource allocation in link layer and the number of sessions. To evaluate the performance of the proposed approach, the tuned CWND was implemented on the TCP Linux kernel. We conducted an experiment on our test-bed to compare the performance of the proposed TCP, TCP-split connection and an E2E TCP variants. As the results indicate, the proposed scheme maintains unreduced CWND in high BER over the satellite link. Similarly, it has higher throughput when comparing TCP variants and connection splitting based PEP solution in both single and multiple TCP sessions in presence of high link errors. This scheme tries to explore the available resource in TCP. The cross layering between TCP and the link layer is contributing to utilize the available resource even in high BER. The end user's terminals are not required to modify or deploy new TCP protocols due to the enhancing of pair TCP splitting based PEP solution architectures at the end of satellite legs (gateway and satellite terminal).

Acknowledgment

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and future Planning (NRF-2014R1A2A2A01002321) and National GNSS Research Center program of Defense Acquisition Program Administration and Agency for Defense Development.

References

[1] ETSI, "Digital Video Broadcasting (DVB): Interaction Channel for Satellite Distribution Systems," ETSI EN 301 790 v. 1.5.1, May. 2009.

[2] A. Pirovano and F. Garcia, "A New survey on improving TCP performances over Geostationary Satellite Link," Network and Communication Technologies, vol. 2, no. 1, Jan. 2013.

[3] C. Caini, R. Firrincieli, and D. Lacamera, "PEPsal: A Performance Enhancing Proxy for TCP Satellite Connections," IEEE Aerospace and Electronic Systems Magazine, vol. 22, no. 8, pp. B–9–B–16, 2007.

[4] N. G. Weldegiorgis, K. H. Lee, Y. J Choi, and J. H. Kim, "Testbed and Discussion for PEP in Satellite Communications," in Proc. ICEIC 2015, pp. 28-31, Singapore, Jan. 2015.

[5] M. K. Park and D. G. Oh, "Cross-layer design for improving TCP PEP performance in DVB-RCS2 networks," in Proc. IEEE ICTC, pp. 846-847, 2013.

[6] PEPsal Source Code, available at: http://www.sourceforge.net/projects/pepsal/

[7] DumyNet, available at:http://info.iet.unipi.it/~luigi/dummynet/

[8] Iperf source software, available at: http://software.es.net/iperf/