Adaptive Borrow-Repay Resource Allocation on Disruptive Satellite Communication with PEP

Ji-Hoon Oh
Department of Electrical and Computer Engineering
Ajou University
Suwon, Korea
ggrrzz@ajou.ac.kr

Jong-Mu Kim
Samsung Electronics Co., Ltd.,
Suwon, Korea
blackkim822@gmail.com

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

***Abstract*— Adaptive borrow-repay resource allocation uses the positioning information when temporal communication disconnection is expected. It borrow the required bandwidth and repay later. Obstructions such as tunnels or buildings between satellite terminals and satellites may temporarily disconnect the satellite communication and result in the performance degradation of TCP. In this paper, we propose the novel adaptive borrow-repay resource allocation mechanism using cross-layer information exchange between an application layer and a link layer. The proposed allocation mechanism uses the positioning information of the satellite communication on the move (SOTM) node and the number of sessions. To evaluate the performance, we implemented a test-bed which includes the proposed mechanism. The experimental results show that the performance of the proposed mechanism can improves the TCP throughput and fairness for a temporarily disconnected satellite network user.**

Keywords—Performance Enhancing Proxy, Cross-Layer, Positioning Information, Number of Sessions

I. Introduction

Satellite communication systems such as digital video broadcasting-return channel via satellite (DVB-RCS) represent a viable solution for Internet access in wide areas. Especially, satellite communication on the move (SOTM) has become a significant part of commercial and military communications because they require a high-speed and a world-wide communication service [1]. DVB-RCS is being designed to be fully Internet protocol (IP) based. Therefore, it is an ideal means for offering transmission control protocol (TCP)/IP based Internet services over long distances to achieve a reliable end-to-end (E2E) connection for data delivery.

Originally, TCP was designed for wired networks which have short propagation delay and error-free properties. In satellite networks, however, the performance of conventional TCP protocols are severely degraded because of following three reasons: first, many people share limited bandwidth, so some people cannot get enough bandwidth. Second, satellite communications are characterized by long round trip-time (RTT), so long time is required to increase sufficient congestion window (CWND). Finally, there is a high packet error rate (PER) on the radio link which causes unnecessary congestion control [2].

Nowadays, researches on a TCP splitting performance enhancing proxy (PEP) are performed to enhance the TCP performance in the satellite communication [3]-[5]. TCP splitting PEPs isolate the satellite network from the terrestrial networks in order to exploit the advantages of advanced and customized TCP versions designed for satellite links. However, TCP splitting PEP still has low performance when the PEP experience temporal disconnection due to line of sight obstructions such as tunnels and buildings. These obstructions can temporarily disconnect satellite communication and degrade the performance of TCP.

To solve the problem, we propose an adaptive borrow-repay resource allocation mechanism using a cross-layer information exchange between application layer and link layer. In the proposed mechanism, network control center (NCC) allocates more resource to SOTM node before it enters the tunnel to improve TCP throughput. For the adaptive borrow-repay resource allocation mechanism, NCC uses the positioning information of SOTM node and the number of sessions. The performance of adaptive borrow-repay resource allocation mechanism is evaluated using the implementation of testbed. In addition, we analyze and compare the fairness of the SOTM nodes.

The remainder of this paper is organized as follows. Section II describes related works. In Section III, a detailed description of the proposed protocol is presented. Section IV explains an architecture of test-bed for the evaluation of proposed mechanism. Section V describes the performance evaluation of the proposed protocol. Finally, the conclusion will be followed.

II. Related Works

A. Cross-Layer Design

Many studies have been performed in cross-layer design, which exchange the information of different layer to enhance the performance of satellite communications [6]-[9]. According to [6], a scheduler has been proposed on the basis of cross-layer design between the physical and link layer medium access control (MAC) layers. In [7], by the exchange of information between the TCP and the link layer, the congestion avoidance method was proposed to prevent overflow in a queue due to the long delay of satellite link. By using the size of the queue in a lower layer, the CWND of TCP is adjusted in [8] and [9].

B. TCP Variants

Due to the characteristics of wireless channel, existing TCPs severely experience performance degradation in satellite networks. To mitigate TCP performance problems in these networks, many solutions have been proposed. One of the solutions is an enhancement of E2E TCP protocols [10]-[15]. Conventional TCP does not fully utilize the network capacity in large bandwidth-delay product (BDP) links due to the unsuitable congestion control algorithm. Many loss-based or delay-based congestion control algorithms have been developed and ongoing to improve the TCP throughput in satellite links. Comparative analysis of several techniques to improve the E2E performance of TCP over lossy wireless links have been studied [13]. They typically involve tuning TCP so that the long RTTs of satellite links do not negatively impact on performance [14]. Two versions of TCP that perform well over high-speed networks are TCP Hybla [10] and TCP CUBIC [11]. TCP Hybla avoids the performance difference that arises from long RTTs. The basic idea is to obtain the same instantaneous segment transmission for the long RTT links (e.g. satellite links). TCP Hybla has many process steps for change of the CWND, which are slow start mechanism, congestion avoidance phases, channel estimation, the selective ACK (SACK) policy and the adoption of a time stamp and segment interval time. CUBIC modifies the window control and improves its TCP friendliness and RTT-fairness. It uses a cubic increase function which derives the elapsed time since the last loss event occurs. These enhanced TCP protocols achieve high throughput even in long propagation delay. However, they decrease performance aggressively as the transmission error increases [2], [13], [14]. Advanced satellite transport protocol (ASTP) has been designed to adapt satellite characteristics [15]. ASTP attains high throughput and exploits the knowledge of the bandwidth allocated to each terminal, where the knowledge is available from the satellite network operator. It is already built the concept of integrated injection control theory in active queue management (AQM) control. Due to the specialized nature of the protocol, however, every node on the network needs to modify its protocol. In addition, these protocol has some issues related to fairness with other TCP variants.

III. PROPOSED PROTOCOL

A. Proposed System Model

The system model defined in this paper is shown in Fig. 1. It consists of a SOTM node, satellite, ground station and file server. SOTM node and ground station are connected via satellite link. Ground station and file server are connected via wired link. TCP splitting PEP is installed in ground station and PEP divides TCP connection. Wired link uses the TCP Reno protocol to ensure compatibility with the file server and the ground station. Satellite link uses the TCP Hybla protocol to ensure compatibility with the SOTM node and the ground station.

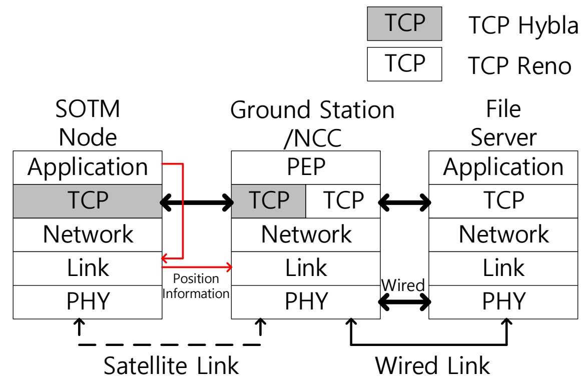

Figure 1. System model of the proposed protocol

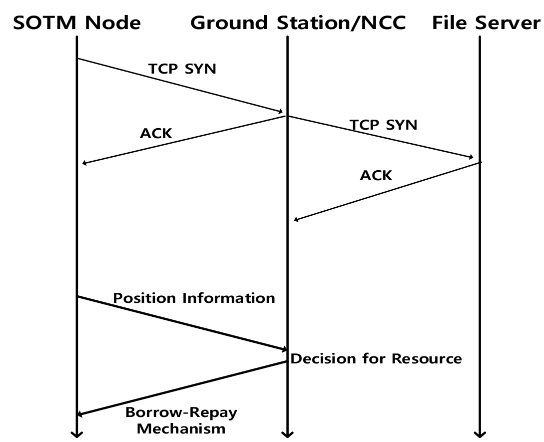

Figure 2. PEP initiation and position information transmission procedure

A cross-layer design allows direct data exchange between non-adjacent layers. The link layer is logically situated above the physical layer and it delivers data across the satellite link. To set the optimum resource allocation, each session for a satellite link positioning information in application layer is delivered positioning information to link layer. Link layer provides positioning information to NCC.

The procedure of PEP initiation and adaptive borrow-repay resource allocation mechanism is shown in Fig. 2. For the splitting, if ground station gets the TCP SYN between SOTM node and file server, PEP in ground station sends an ACK to SOTM node instead of file server. Then, ground station and file server has TCP connection. Disadvantages of long RTT is avoided by controlling CWND separately in terrestrial and satellite link.

The application layer of the SOTM node sends location information to the DAMA agent at the link layer. SOTM node sends the request message of resource allocation including location information and protocol overheads to the DAMA controller of the NCC. The resource allocation is determined by the DAMA controller in ground station based on the channel conditions, available resources, and location information of SOTM node. The DAMA controller in ground

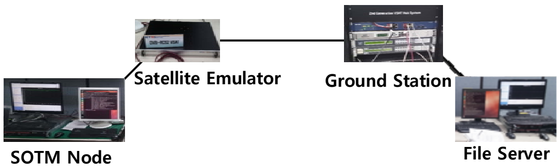

Figure 3. PEP testbed architecture

Table 1. Testbed environment

Testbed components	Detailed description
SOTM Node	OS: Linux (Kernel 3.17.4) 1 Giga LAN interface Iperf (Version 3.0.6)
Satellite Emulator	OS: FreeBSD 2 Giga LAN interfaces DummyNet
Ground Station	OS: Linux (Kernel 3.17.4) 2 Giga LAN interfaces PEPsal (Version 2.0.1)
File Server	OS: Linux (Kernel 3.17.4) 1 Giga LAN interface Iperf (Version 3.0.6)

station allocates resources for the DAMA agent and report it to the DAMA agent through TBTP message on the forward link.

B. Proposed Adaptive Borrow-Repay Resources Allocation

The performance of a SOTM node is degraded in the tunnel, because communication between SOTM nodes and satellite is temporarily disconnected. To reduce impact of performance degradation, we propose an adaptive borrow-repay resource allocation mechanism using positioning information. In the proposed mechanism, if the SOTM node has recognized that it will enter a tunnel in some time using the positioning information, it sends positioning information to NCC. NCC allocates more resource to improve TCP throughput based on SOTM node positioning information. When SOTM node enters the tunnel, NCC compensates resource which is allocated before enters the tunnel.

NCC need to calculate additional resources to be allocated to the SOTM node. First, SOTM node calculate additional resources that cannot be used due to temporal communication disconnect. An amount of unused resource D_u is calculated using

$$D_u = R \times T_t , \quad (1)$$

where R and T_t is resource allocation of SOTM node and the time in the tunnel.

Second, we need to get how many resources should be borrowed for each session D_b. We use

$$D_b = \frac{R \times T_t}{T_r \times (N-1)} , \quad (2)$$

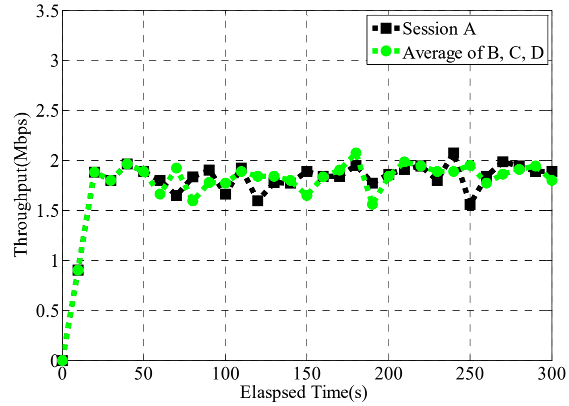

Figure 4. TCP throughput without temporal disconnection

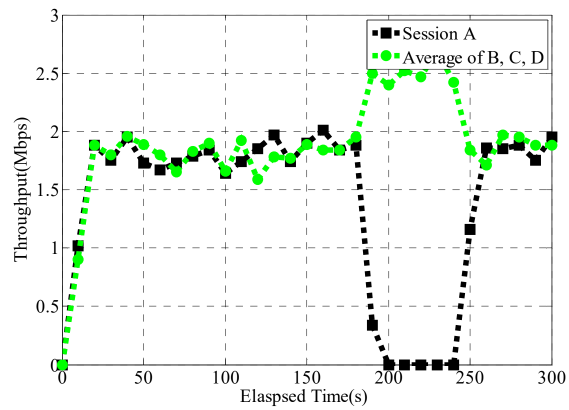

Figure 5. TCP throughput with temporal disconnection

where N is number of sessions and T_r is the difference between the time when SOTM node actually enters tunnel and the time when it recognizes that it enters the tunnel.

When SOTM node enters the tunnel, it repays the borrowed resource which is allocated to SOTM node before it enter the tunnel. Amount of repay resource D_r is calculated by

$$D_r = \frac{R \times T_t}{N-1} . \quad (3)$$

IV. Implementation of Testbed

We implemented a test-bed for the evaluation of the proposed mechanism as shown in Fig. 4. Testbed environment is shown in Table 1. Test-bed consists of 4 devices, which are corresponding to SOTM nodes, a satellite emulator, a ground station and a file server. The proposed adaptive borrow-repay resource allocation mechanism is installed in the ground station. The third box from the left end in Fig. 3 represents ground station and NCC. We installed PEPsal [16] on these box to divide E2E TCP connections into two segments. PEPsal is an open source software for Linux operating system (OS). This allows to enable adaptive borrow-repay resource allocation on the satellite link. It also uses Netfilter [17] to intercept incoming segments and to copy them into a queue. The queue

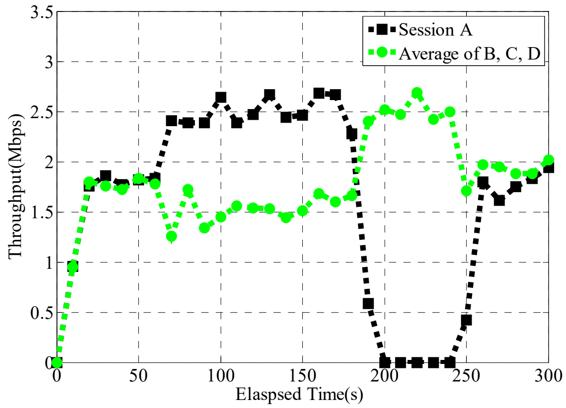

Figure 6. TCP throughput of proposed mechanism with temporal disconnection

Table 2. Transmitted data of each session for 300s

Session	Transmitted data (Mbits)		
	Without temporal disconnection	**With temporal disconnection**	**Proposed mechanism with temporal disconnection**
A	543.2	427.2	**498.4**
B	544.8	575.2	**544.8**
C	543.2	577.6	**543.2**
D	542.4	576.8	**544**

handler classifies the information (IP addresses and TCP ports) on the two end users.

The satellite emulator in Fig. 3 emulates the wireless satellite link using Dummynet [18]. Dummynet simulates bandwidth limitations, delays, packet losses and multipath effects. It is operated on FreeBSD OS. To measure network performance in our testbed, we install iperf [19] on both the end nodes. Iperf measures the throughput of a network.

V. Performance Evaluation

This section presents the performance results obtained from experiments in various scenarios. We assumed that the SOTM node is on a highway with/without a tunnel (satellite channel block environment). We compared the 3 cases of performance of TCP throughput: 1) conventional TCP without communication disconnection, 2) conventional TCP with temporal communication disconnection, and 3) Proposed mechanism with temporal communication disconnection. SOTM node enters the tunnel after 3 minutes from starting the TCP communication. SOTM node stays in the tunnel for 1 minute. SOTM node recognizes that it entering the tunnel using the positioning information before 2 minutes ago from entering. The number of sessions are four.

The satellite channel has limited bandwidth of 8 Mbps. RTT is set as 500 ms and the number of segments assigned to the congestion parameter for TCP in Linux kernel is set as 1,448 bytes.

Table 3. Result of TCP fairness

Number of Sessions	Jain's Fairness Index	
	TCP split PEP	**Proposed mechanism**
2	0.962	**0.99**
4	0.976	**0.997**
8	0.987	**0.999**

A. *Conventional TCP without Temporal Disconnection*

First, we evaluate the conventional TCP performance without temporal disconnection network for an elapsed time of 5 minutes over error-free channel condition. As in Fig. 4, TCP throughputs of each session have almost the same throughput.

B. *Conventional TCP with Temporal Disconnection*

Second, we evaluate conventional TCP performance with temporal disconnection network for an elapsed time of 5 minutes over error-free channel condition. A SOTM session A enters the tunnel when elapsed 3 minutes. As in Fig. 5, session A throughput critically degraded when elapsed 3 minutes after enter the tunnel because line of sight obstruction is caused by tunnel.

C. *Proposed Mechanism with Temporal Disconnection*

Third, we evaluate TCP performance with temporal disconnection network using adaptive borrow-repay resource allocation mechanism for an elapsed time of 5 minutes over error-free channel condition. After 1 minute from starting the TCP communication, i.e. before the tunnel entrance 2 minutes ago, session A recognizes the entering the tunnel. Session A sends the positioning information to the NCC. NCC, in ground station using positioning information of session A, NCC allocates more resource. When session A enters the tunnel, Session A repay B, C, D to pay off the resources that have been borrowed by distributing. As in Fig. 6, SOTM node of session A achieves higher throughput. The results of transmitted data are shown table 2. The table implies proposed mechanism is fairer than existing scheme.

D. *Fairness of the Network*

Finally, we analyzed the fairness of the network in a temporal disconnection network for an elapsed time of 5 minutes over the error-free channel condition. The fairness of the network is determined by the Jain's fairness index as follows [20].

$$f = \frac{(\sum_{i=1}^{n} t_i)^2}{N\sum_{i=1}^{n} (t_i)^2}. \tag{4}$$

Where f is the fairness index value, N is the number of nodes, t_i is the throughput of the node i. The fairness index is between 0 and 1. The results of fairness index are shown in Table 3. The table implies that the adaptive borrow-repay resource allocation mechanism is fairer than conventional TCP.

VI. CONCLUSION

In this paper, we proposed a novel adaptive borrow-repay resource allocation mechanism using positioning information of SOTM node and the number of sessions. We measured TCP throughput and fairness on the test-bed to compare the performance of conventional TCP without temporal disconnection case, conventional TCP with temporal disconnection case and proposed adaptive borrow-repay resource allocation mechanism with temporal disconnection. As the results indicate, the proposed adaptive borrow-repay resource allocation mechanism achieves higher throughput over the satellite link with temporal disconnection. Similarly, it has higher throughput comparing to without adaptive borrow-repay resource allocation mechanism. The cross-layering between application layer and the link layer is contributing to utilize the available resource even in temporal disconnection of satellite communication.

ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. NRF-2017R1A2A2A05001404).

REFERENCES

[1] W. M. Smith, “Channel characterization and modeling for satellite communications on the move”, in Proc. IEEE MILCOM, (2005), pp. 821–827

[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, D. G. Oh, Y. J. Choi, and J. H. Kim, “Building Three - segment splitting PEPsal Testbed and Performance Evaluation,” in Proc. Proce edings of Symposium of the Korean Institute of communications and Information Sciences, pp. 998-999, Jan. 2015.

[6] E. Rendon-Morales, J. Mata-Diaz, J. Alins, J. L. Munoz, and O. Esparza, “Cross-layer Packet Scheduler for QoS Support over Digital Video Broadcasting-Second Generation Broadband Satellite Systems,” International Journal of Communication Systems, pp. 1–20, 2012.

[7] F. Peng, L. Wu, and V. C. M. Leung, “Cross-layer Enhancement of TCP Split-connections over Satellites Links,” International Journal of Satellite Communications and Networking, vol. 24, no. 5, pp. 405–418, 2006.

[8] M. Luglio, F. Zampognaro, T. Morell, and F. Vieira, “Joint DAMATCP Protocol Optimization Through Multiple Cross Layer Interactions in DVB RCS Scenario,” in Proc. IEEE IWSSC 2007, Sep. 2007.

[9] “XPLIT: A Cross-layer Architecture for TCP Services over DVBS2/ETSI QoS BSM,” Computer Networks, vol. 56, no. 1, pp. 412–434, 2012.

[10] C. Caini and R. Firrincieli, “TCP Hybla: A TCP Enhancement for Heterogeneous Networks,” International Journal of Satellite Communications and Networking, vol. 22, no. 5, 2004.

[11] S. Ha, I. Rhee and L. Xu, “CUBIC: a new TCP-friendly high-speed TCP variant,” in Proc. SIGOPS Oper. Syst. Rev., vol. 42, pp. 64–74, July 2008.

[12] R.H. Wang and S. Horan, "Performance evaluation of TCP and its extensions over lossy links in a small satellite environment," in Proc. ICC 2005, Vol. 3, 16-20 May 2005

[13] J.V. Maisuria and R.M. Patel, "Overview of Techniques for Improving QoS of TCP over Wireless Links," in Proc. CSNT 2012, 11-13 May 2012.

[14] S. Trivedi, S. Jaiswal, R. Kumar, and S. Rao, "Comparative performance evaluation of TCP Hybla and TCP Cubic for satellite communication under low error conditions," in Proc. IMSAA 2010, 15-17 Dec. 2010.

[15] M. Muhammad, F. Kasmis, and T. de Cola, “Advanced Transport Satellite Protocol,” in Proc. GLOBECOM 2012, Anaheim, CA, USA, 2012.

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

[17] The Netfilter/iptables project, firewalling, NAT and packet mangling for Linux, available at: http://www.netfilter.org.

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

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

[20] R. Jain, The Art of Computer Systems Performance Evaluation, Wiley, New York, 1991.