TCP ADaLR+: Enhanced TCP Scheme for GEO Satellite Networks

Liang Zong[1,2], Wencai Du[1*], Yong Bai[1]

[1]*College of Information Science & Technology, Hainan University, Haikou, Hainan 570228, China*

[2]*Department of Information Science and Engineering, Hunan University of Humanities, Science and Technology, Loudi, 417000, China*

Received 1 October 2014, www.cmnt.lv

Abstract

TCP performance is essential for data transmissions over the satellite network. The TCP ADaLR is congestion control algorithm that the sender judgments the relevant window change and measures round-trip time to control congestion window. It can adapt to the characteristics of the satellite link and improve the performance of TCP than conventional TCP (New Reno). However, it doesn't take into account distinction of random packet loss and congestion packet loss like the TCP Veno. In this paper, we propose further enhancement scheme of TCP ADaLR, called TCP ADaLR+, which can distinguish between random packet loss and congestion loss. The improved performance of proposed TCP ADaLR+ is demonstrated by simulations. In all simulation scenarios, TCP ADaLR+ outperforms TCP ADaLR and TCP Veno in terms of satellite link throughput and FTP download response time.

Keywords: satellite networks, TCP ADaLR, random loss

1 Introduction

With the rapid development of the Internet and the mobile Internet, the existing computer networks increasingly transition from the wired networks to the wireless networks. However, there are a lot of geographical areas uncover by the terrestrial infrastructure, such as the ocean, military battlefield, and remote rural areas. The satellite networks may become the only choice for mobile users at sea to connect with people on land. However, TCP performance degrades in geostationary satellite networks due to long propagation delays and high bit error rates. In recent years, many of scholars propose a number of solutions for improve TCP performance in satellite networks [1-4]. The proposed solutions can be roughly divided into the following three categories: end-to-end improvement solutions [5], TCP **s**plit connection solutions [6] and delay tolerant networks (DTN) solutions [23].

End-to-end improvement solutions which can compatible with the existing network protocols just need to modify the end nodes. Tropea et al. proposed an available bandwidth adaptable Transport Protocol (ATP) and it consists of a Satellite Transport Protocol (STP) protocol with the addition of the TCP Jersey bandwidth estimation algorithm called available bandwidth estimation (ABE) [7]. Luglio et al. proposed a satellite-based architecture in which DVB-S2 and DVB-RCS standards, MIPv6 at layer 3 (for handover management), and an enhanced TCP version at layer 4, named TCP Noordwijk, allow to achieve good performance [8]. Nguyen et al. evaluated the overall performance of AeroTP by comparing it with TCP New Reno and Westwood [9]. Muhammad et al. proposed a novel end-to-end (E2E) transport layer protocol, namely Advanced Transport Satellite Protocol (ATSP), which is built around consolidated control theory concepts already infused in Active Queue Management (AQM) control schemes. ATSP exploits the knowledge of the bandwidth allocated to each terminal, as available from the satellite network operator [10].

Peng et al. evaluated and compared popularly used TCP versions such as New Reno, Hybla, Vegas, DVegas and Westwood+ over GEO VSAT and LEO-based satellite links employing performance enhancement proxy (PEP) based on snoop [11]. Park et al. proposed ACKs-first variable-size queuing (AFVQ) for a gateway and derived an analytic model of the steady-state TCP performance with bidirectional traffic to clearly identify the two sources of the problem: the excessive queuing delay of ACK packets and the excessive number of ACK packets in the queue [12]. Sacchi et al. designed an innovative physical (PHY) layer for broadband satellite connections operating in W-band, which was based on the prolate spheroidal wave functions (PSWFs) [13]. Kronewitter et al. proposed a novel solution, namely broadband HAIPE-embeddable satellite communications terminal (BHeST), which utilizes dynamic network performance enhancement algorithms for high latency bandwidth-on-demand satellite links protected by a high assurance IP encryption (HAIPE) [14]. Wang et al. presented a survey on the protocols proposed for reliable data transport in space Internet, with a focus on the latest developments [15]. Celandroni et al. presented the reference architecture, the access scheme, the choice of the most suitable transmission parameters and the simulation results that are obtained when transmitting the two different types of data [16].

TCP split connection solutions have two methods: TCP-Spoofing [17] and TCP-Splitting [18,19]. The

* ***Corresponding author's*** e-mail: wencai@hainu.edu.cn

difference is that TCP-Spoofing maintaining the integrity of the TCP connection; TCP-Splitting put a TCP connection into multiple independent TCP connections.

Dubois et al. focused on mobility scenarios and proposed some architecture elements on PEPs mechanisms for satellite networks [20]. Pirovano and Garcia proposed an overview of TCP variants and a survey of commonly proposed solutions for TCP over satellite and assessed the gain offered by a split TCP connection [21]. Jorge et al. proposed the split TCP and UDP, which splits the TCP connection and uses a customized and lighter transport protocol for the wireless segment, takes advantage of the IEEE 802.11e Hybrid Coordination Function Controlled Channel Access (HCCA) mechanisms to remove redundant TCP functionalities [22].

Caini et al. discussed the advantages and disadvantages of the DTN approach compared to the more conventional solutions and adopted the DTN architecture based on the introduction of the new "bundle" layer in the protocol stack [23]. Caini et al. compared the DTN and PEP architectures and stacks, as a preliminary step for the subsequent DTN performance assessment carried out in practical LEO/GEO satellite scenarios [24]. Yu et al. presented an experimental investigation of the custody transfer on communications characterized by the very long signal propagation delay, lengthy link disruptions and high data-loss rates that are typical of deep space links [25]. Celandroni et al focused on a set of selected satellite-based scenarios, where the DTN paradigm adoption greatly improves the unicast and multicast communications' performance [26].

From the existing research results, the studies combine the long delay and high bit error rate are few. The PEPs and DTN are hotspot recently, but these change the end-to-end semantics of transport protocol and affect the safety of transmission. This paper proposed the ADaLR+ just modifies the TCP sender and maintains the end-to-end semantics. In addition to the security, the ADaLR+ can improve the utilization rate of the satellite link with congestion environment. Due to the cost of satellite communications is very expensive, and people can greatly reduce the cost of satellite communication. The security of data transmission can protect effectively.

The paper is organized as follows. In Section 2, the TCP-ADaLR algorithm and TCP Veno are analysed. In Section 3, the TCP ADaLR+ proposal is introduced. Simulation results that compare the performance of TCP ADaLR+, TCP ADaLR, TCP Veno and TCP NewReno are presented in Section 4 and conclusions are drawn in Section 5.

2 TCP ADaLR

TCP ADaLR (TCP with algorithm modifications for adaptive delay and loss response) [27, 28] is an end-to-end congestion control algorithm that improves performance for broadband geostationary satellite networks. It introduces division of congestion window *cwnd* increment phase into sub-phases in order to enable transmission of additional segments for better satellite link utilization in the absence of losses. It also adjusts transmission rate more adaptively in the presence of losses. It controls congestion window using the scaling factor ρ which is computed from the measured RTT. It just needs to modify the TCP sender and divides into three processes: adaptive *cwnd* (congestion window) increase mechanism, adaptive *rwnd* (receive window) increase mechanism, and loss recovery mechanism.

TCP ADaLR firstly calculates the scaling factor ρ by the measured RTT:

$$\rho = (\text{sample RTT s}/1\text{ s}) \times 60 \tag{1}$$

The sample RTT is the measured RTT of a data segment sample not retransmitted. The fixed parameter 60 is the minimum recommended value for the maximum RTO. The lower bound and upper bound of ρ is 1 and 60 respectively. A breakpoint $\rho=15$ was selected to define the transition for the adaptive *cwnd* mechanism. According to whether or not the packet loss and the value of ρ, TCP ADaLR adjusts the size of congestion window in the slow start and congestion avoidance phase.

Slow start phase:

// if no losses have occurred and the value of $\rho \geq 15$, the SMSS is the send maximum segment Size

$$\text{CWND}=\left(\frac{\sqrt{\rho}}{4}\right)\times \text{SMSS}.$$

Congestion avoidance phase:

// if no losses have occurred, the value of $\rho \geq 15$, and TCP sender is out of fast or flightsize is less than (rwnd/2)

$$\text{CWND}=\left(\frac{\sqrt{\rho}}{2}/2\right)\times \text{SMSS}\times \text{SMSS}/cwnd$$

In the process of adaptive *rwnd* increase mechanism, the number of the sender transmitted bytes is the minimum of either the *cwnd* or *rwnd*. In order to improve the efficiency of the long delay satellite links, in addition to the adaptive *cwnd* increase mechanism used at the sender, adaptive *rwnd* increase mechanism is used at the receiver. When no losses have occurred, TCP ADaLR allows the transmission of additional segments thus improving bandwidth utilization over the satellite link.

When multiple losses have occurred in fast recovery phase, RWND may increase a SMSS:

// if no losses have occurred

$$\text{RWND}=\text{RWND}+\text{rtt\_dev\_gain}\times \rho \times \text{SMSS}$$

// if losses have occurred and in fast recovery phase

$$\text{RWND}=\text{RWND}+\text{SMSS}.$$

The rtt_dev_gain is deviation gain, set 0.25. The loss recovery mechanism modifies the fast recovery phase to enable the back-to-back transmission of two segments. The maximum ACK delay of 200ms is added to the current time used in the computation of the subsequent RTO value, if ACK delay option is enable. Both modifications compensate for additional delays in sending an ACK if two back-to-back segments are not received by the TCP receiver when the delayed ACK option is enabled.

The loss recovery mechanism limits the number of retransmissions from the retransmission buffer to three segments to prevent a large number of unnecessary or spurious retransmissions.

// if CWND<= acked_bytes

CWND=2×SMSS

// else

CWND=CWND-acked_bytes+(2×SMSS)

TCP ADaLR greatly improves the TCP performance by adaptive *cwnd* increase mechanism; adaptive *rwnd* increase mechanism and loss recovery mechanism. Summary, the adaptive *cwnd* mechanism was proposed to divide the slow start phase into four sub-phases based on the current *cwnd*, slow start threshold (ssthresh), and flightsize (total unacknowledged byte in the network). The *cwnd* increment factor depends on the sub-phases. In the process of adaptive *rwnd* increase mechanism, at the TCP receiver, the *rwnd* limits the amount of transmitted data. The number of transmitted bytes is the minimum of either the *cwnd* or *rwnd*. The adaptive *rwnd* increase mechanism is used to increment the *rwnd* when no losses have occurred. TCP ADaLR allows the transmission of additional segments thus improving bandwidth utilization of a satellite link. The loss recovery mechanism modifies the fast recovery phase to enable the back-to-back transmission of two segments. It limits the number of retransmissions from the retransmission buffer to three segments to prevent a large number of unnecessary retransmissions.

3 TCP ADaLR+

3.1 SLOW START

The TCP ADaLR increases congestion window with different values based on the window and the segment number in the slow start and congestion avoidance. It is not taking into account the random packet loss in the wireless environment. The sender considers that all packet loss is caused by congestion and triggers frequently slow start. Therefor the TCP performance dramatically decline, particularly in the long delay and high BER satellite links.

The ADaLR+ improves mainly three aspects: the loss detection, the congestion window increase and the threshold adjustment. It checks the packet loss in the congestion avoidance and fast recovery phase. Thus it distinguishes effectively the random loss and the congestion loss, and minimizes the impact of random packet loss for the TCP performance.

When a packet loss is detected, if $N \geq \beta$, the backlog N at the queue is large, then ADaLR+ treats the packet loss as the congestion loss. The value of congestion window can take the value of the traditional TCP congestion window. If $N<\beta$, the backlog N at the queue is small. ADaLR+ treats the packet loss as a random loss. The value of the congestion window can take the value of the TCP Veno. The TCP ADaLR divided the slow start into four different phases. However, all of the congestion window are used CWND=($\sqrt{\rho}/4$)×SMSS in every phase, which does not reflect the difference.

In the four sub-phases, the TCP ADaLR+ has a different increase for the congestion window in order to make it adapted to the different growth in the slow start. The slow start begins in the initial connection or the timeout. The sender need detect the network state in the first phase when the initial connection. The value of the congestion window cannot be too high. The ADaLR+ still uses the TCP ADaLR's settings. The CWND=($\sqrt{\rho}/4$)×SMSS. In the second phase and the third phase, the rate of window increase can increase substantially. The ADaLR+ sets the CWND is $\sqrt{\rho}$ ×SMSS. The last phase may be appropriate to reduce the rate of window increase, and also used the TCP ADaLR's settings, as shown in Figure 1.

3.2 DISTINGUISH THE RANDOM LOSS

The reliability of the wireless link is not stable enough in a wireless network environment. This will often result in the higher bit error rate. The network may cause randomly packet loss, especially in long delay satellite link which BER can be as high as 10^{-4}.

If the network takes the traditional TCP, such as TCP Reno, will treat random packet loss as congestion loss, which will results in the TCP performance degradation. TCP Veno [29] is a performance enhancement for TCP Reno in the wireless environment. It can distinguish effectively between random packet loss and congestion loss in the wireless environment, and includes four phases: slow start, congestion avoidance, retransmission timeout, fast retransmit and fast recovery. Based on the RTT and congestion window, TCP Veno distinguishes between the congestion loss and the random loss.

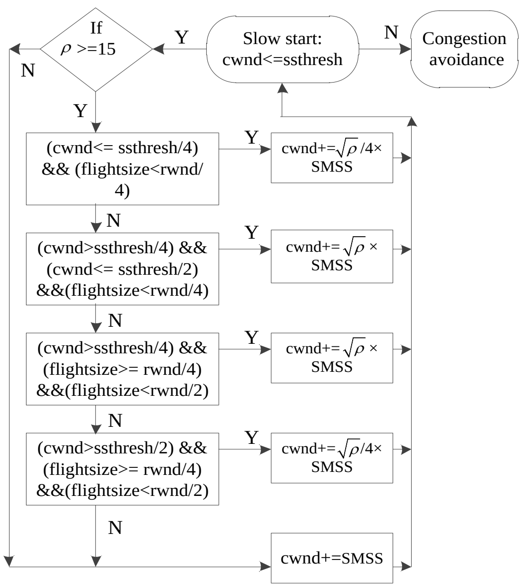

FIGURE 1 Four sub-phases in the slow start

Expected rate: EXPECTED=*cwnd*/BASERTT.

Actual rate: ACTUAL=*cwnd*/RTT.

The RTT is the smoothed round-trip time measured. The BASERTT is the minimum of measured round-trip time. The difference of the rate is:

DIFF=EXPECTED-ACTUAL.

Let N is the backlog at the queue:

N=ACTUAL×(RTT-BASERTT)=DIFF×BASERTT.

When the packet loss is detected, if $N<\beta$, TCP Veno treats the packet loss as a random packet loss; if $N\geq\beta$, as congestion loss. The value of β sets 3.

Fast recovery phase: the ADaLR also does not consider the distinction between the random loss and the congestion loss when sets the threshold in the fast recovery. It also reduces the window by half as same to the traditional TCP (New Reno). So it also greatly reduces the TCP performance. The ADaLR+ modifies the TCP Veno settings.

The value of β set 3 to distinguish between random loss and congestion loss is not appropriate for the long delay and high BER satellite link. ADaLR+ takes three phases, as shown in Figure 2. When N<=3, the threshold takes Veno setting, is 4/5*cwnd*. Then two intervals are increased by ADaLR+ for the threshold setting, when 3<=N<6, the threshold set 3/5*cwnd*, the final interval of the threshold is set 2/5cwnd. Although the last threshold value is smaller than traditional fast recovery set 1/2*cwnd*, considering ADaLR+'s the initial congestion window is larger. The threshold is still high value relatively. The different threshold is set by the value of backlog that makes ADaLR+ have multiple differentiations. In the case of high bit error rate satellite link, ADaLR+ threshold adjustment is more reasonable. The multi-level differentiation also can better adapt to the characteristics of the long delay and satellite different network status.

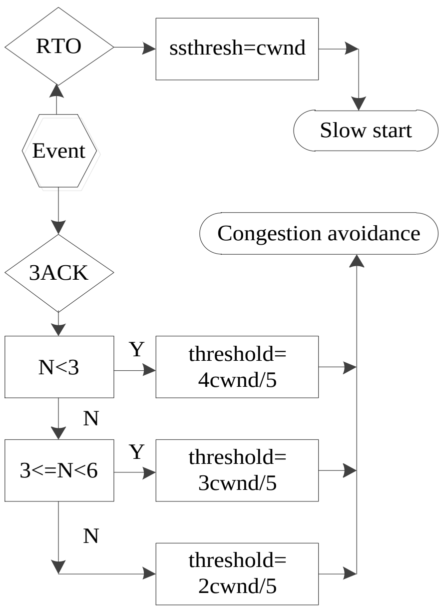

FIGURE 2 Threshold setting in fast recovery

In summary, the TCP ADaLR+ distinguishes the satellite link random loss and adjusts the congestion window and the threshold. The TCP ADaLR+ is an enhancement for the TCP ADaLR to improve the TCP performance in the satellite link.

4 Simulation and results

The simulation is conducted using OPNET software. The basic model is shown in Figure 3. The server is sender on the ground, and connects to the gateway. The gateway sends the data to the satellite which connects to the client. The link rate of the server connect to the gateway is set to 10Mb/s. The downlink (gateway to client) and uplink (client to gateway) data rates are 2,048Kb/s and 256Kb/s, respectively. The satellite links one-way propagation delay is 250ms. The BER of the satellite link sets 10^{-9} to 10^{-5}. The server sends a 50M file size to the client. We evaluate and compare the performance of TCP ADaLR+, TCP ADaLR, TCP Veno, and TCP New Reno by employing two simulation scenarios: the gateway with congestion loss and without congestion loss. The simulation collects the client download response time and the satellite link throughput for various BER (10^{-9} to 10^{-5}).

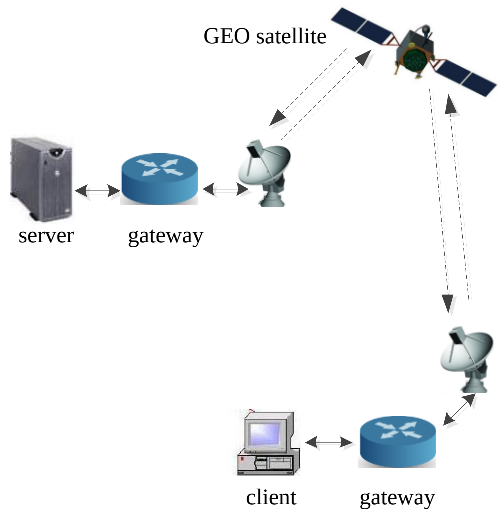

FIGURE 3 Simulation model

In the simulation scenario that the gateway without congestion loss, for illustration, Figure 4 shows the response time of the four TCP variants for various BERs (10^{-9} to 10^{-5}). With the BER increases, the response time increases gradually. Especially the BER changes from 10^{-6} to 10^{-5}, the download response time rapidly increases. In the process of the BER changes from 10^{-9} to 10^{-6}, the response time of the four TCP variants changes between 0.5×10^3s to 2×10^3s, and the difference for the four TCP variants is not great.

The TCP Veno lower than the TCP New Reno when BER is 10^{-5}. The TCP ADaLR+ also slightly lower than the TCP ADaLR. TCP ADaLR+ exhibits the best performance in the four TCP variants.

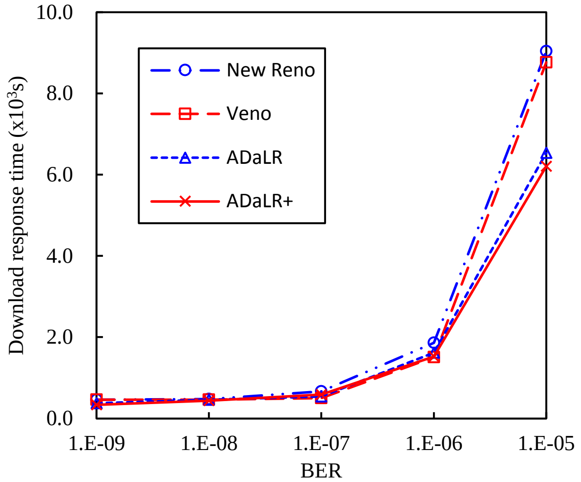

FIGURE 4 Download response time for the scenario without congestion loss

In the simulation scenario that the gateway without congestion loss, for illustration, Figure 5 shows the satellite link throughput of the four TCP variants for various BERs (10^{-9} to 10^{-5}). With the BER increases, the satellite link throughput decreases gradually. Especially the BER changes from 10^{-7} to 10^{-5}, the throughput rapidly decreases.

At higher BERs (10^{-5}), the TCP ADaLR+'s satellite link throughput exhibits better performance than other TCP variants. It shows up nearly 20% higher satellite link throughput than TCP ADaLR. When there is no congestion in the gateway, ADaLR+'s the satellite link throughput higher than ADaLR. The TCP ADaLR's satellite downlink utilization changes from 70% (BER is 10^{-9}) to 13% (BER is 10^{-6}). The TCP ADaLR+'s satellite downlink utilization changes from 76% (BER is10^{-9}) to 16% (BER is 10^{-6}).

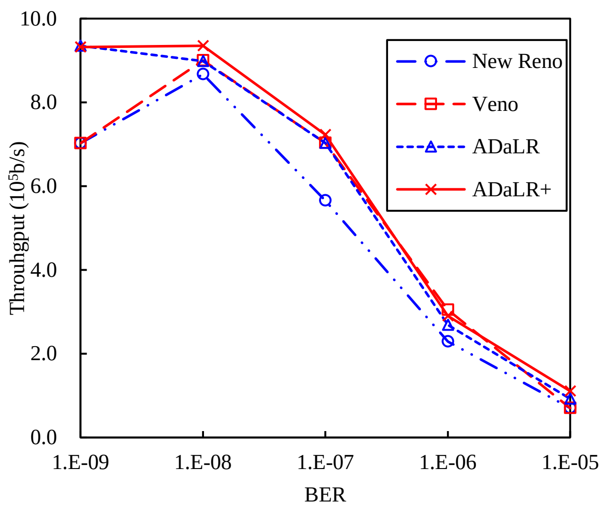

FIGURE 5 Satellite link throughput for the scenario without congestion loss

In the simulation scenario that the gateway with congestion loss, for illustration, Figure 6 shows the response time of the four TCP variants for various BERs (10^{-9} to 10^{-5}). With the BER increases, the response time also increases gradually. Especially the BER changes from 10^{-6} to 10^{-5}, the download response time rapidly increases. In the process of the BER changes from 10^{-9} to 10^{-6}, the response time of the four TCP variants changes between $1.2x10^3$ s to $2.3x10^3$s, and the difference for the four TCP variants is not great.

The response time of the TCP New Reno is longest. The TCP Veno lower than the TCP New Reno when BER is 10^{-5}. The TCP ADaLR+ also is lower than the TCP ADaLR. At lower BERs (10^{-9} to 10^{-6}), the TCP ADaLR and the TCP ADaLR+ exhibit comparable performance to the TCP Veno and the TCP NewReno because the packet losses are mainly due to congestion. At higher BERs (10^{-6} to 10^{-5}), the TCP ADaLR+ exhibits better performance than other TCP because the packet losses are mainly due to the random loss. It can distinguish effectively the ransom loss and the congestion loss and improve TCP performance. The TCP ADaLR+ exhibits up to 18% decrease in download response times than the TCP ADaLR.

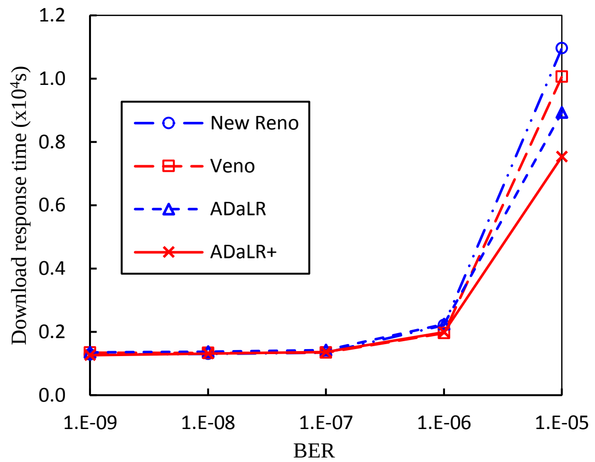

FIGURE 6 Download response time for the scenario with congestion loss

In the simulation scenario that the gateway with congestion loss, for illustration, Figure 7 shows the satellite link throughput of the four TCP variants decreases gradually for various BERs (10^{-9} to 10^{-5}).

At higher BER (10^{-5}), the TCP Veno's satellite link throughput exhibits better performance than TCP New Reno. The TCP ADaLR+'s performance is best. It shows up nearly 13% higher satellite link throughput than TCP ADaLR. The TCP ADaLR's satellite downlink utilization changes from 48% (BER is 10^{-9}) to 12% (BER is 10^{-6}).The TCP ADaLR+'s satellite downlink utilization changes from 55% (BER is 10^{-9}) to 15% (BER is 10^{-6}).

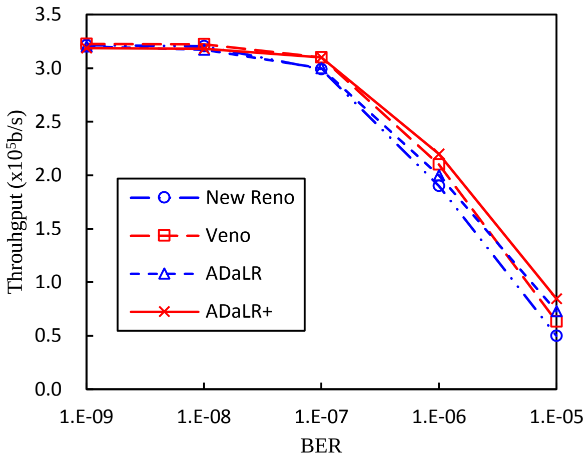

FIGURE 7 Satellite link throughput for the scenario with congestion loss

5 Conclusion

In this paper, we presented an enhanced TCP scheme, TCP ADaLR+, to take the advantages of TCP ADaLR and TCP Veno. TCP ADaLR+ can distinguish the random loss and congestion loss, and modify the slow start phase with different slow start threshold settings for these two different kinds of losses. The simulation results show that TCP ADaLR+ exhibits better performance than other TCP variants (TCP ADaLR, TCP Veno, and TCP New Reno) for communications over satellite network. The TCP ADaLR+ does not violate the end-to-end semantics of TCP, and requires modifications only at the TCP sender.

Acknowledgments

This paper was supported by the National Natural Science Foundation of China under Grant No.61162010 and No.61261024 and Hunan Provincial Natural Science Foundation of Chin (12JJ2040).

References

[1] Allman M, Glover D, Sanchez L, Enhancing 1999 TCP over satellite links using standard mechanisms *RFC* 2488

[2] Allman M, Dawkins S, Glover D 2000 Ongoing TCP research related to satellites *RFC* 2760

[3] Taleb T, Hadjadj-Aoul Y, Ahmed T 2011 *IEEE Wireless Communications* **18**(1) 146-52

[4] Botta A, Pescapé A 2014 *IEEE Communications Magazine* **52**(6) 201-9

[5] Biswas I 2013 An investigation on TCP large initial window *International Journal of Satellite Communications and Networking* **31**(3) 111-21

[6] Pirovano A, Garcia F 2013 A New Survey on Improving TCP Performances over Geostationary Satellite Link *Network & Communication Technologies* **2**(1) 1-18

[7] Tropea M, De Rango F, Fazio P, et al 2013 A new TCP Bandwidth Estimation protocol for satellite networks *9th International IEEE Wireless Communications and Mobile Computing Conference (IWCMC)* 13-8

[8] Luglio M, Roseti C, Zampognaro F 2012 Transport layer enhancements on a satellite - based mobile broadband system for high - speed trains *International Journal of Satellite Communications and Networking* **30**(6) 235-49

[9] Nguyen T A N, Gangadhar S, Umapathi G, et al 2012 Performance Evaluation of the AeroTP Protocol in Comparison to TCP NewReno, TCP Westwood and SCPS-TP *Proceedings of the International Telemetering Conference (ITC)*

[10] Muhammad M, Kasmis F, De Cola T 2012 Advanced transport satellite protocol *IEEE Global Communications Conference (GLOBECOM)* 3299-304

[11] Peng F, Cardona A S, Shafiee K, et al 2012 TCP Performance Evaluation over GEO and LEO Satellite Links between Performance Enhancement Proxies *IEEE Vehicular Technology Conference (VTC Fall)* 1-5

[12] Park J, Park D, Hong S, et al 2011 Preventing TCP performance interference on asymmetric links using ACKs-first variable-size queuing *Computer Communications* **34**(6) 730-42

[13] Sacchi C, Rossi T, Ruggieri M, et al *IEEE Transactions on Aerospace and Electronic Systems* **47**(2) 974-95

[14] Kronewitter F D, Ryu B, Zhang Z, et al 2011 TCP accelerator for DVB-RCS SATCOM dynamic bandwidth environment with HAIPE *Journal of Communications and Networks* **13**(5) 518-24

[15] Wang R, Taleb T, Jamalipour A 2009 *IEEE Communications Surveys & Tutorials* **11**(2) 21-32

[16] Celandroni N, Davoli F, Ferro E, et al 2010 Medium access control scheme for supporting user mobility in digital video broadcasting-return channel via satellite/satellite second generation-general architecture and functionalities *IET Communications* **4**(13) 1532-43

[17] Ishac J, Allman M 2001 On the performance of TCP spoofing in satellite networks *Military Communications Conference IEEE Communications for Network-Centric Operations: Creating the Information Force* 700-4

[18] Border J, Kojo M, Griner J, Montenegro G, Shelby Z 2001 Performance enhancing proxies intended to integrate link-related degradations *RFC 3135*

[19] Bisio I, Marchese M, Mongelli M 2009 Performance Enhanced Proxy Solutions for Satellite Networks: State of the Art, Protocol Stack and Possible Interfaces *Personal Satellite Services Springer Berlin Heidelberg* 61-7

[20] Dubois E, Fasson J, Donny C, et al 2010 Enhancing tcp based communications in mobile satellite scenarios: TCP peps issues and solutions *IEEE 5th Advanced satellite multimedia systems conference (asma) and the 11th signal processing for space communications workshop (spsc)* 476-83

[21] Pirovano A, Garcia F 2013 A New Survey on Improving TCP Performances over Geostationary Satellite Link *Network & Communication Technologies* **2**(1) 1-18

[22] Navarro-Ortiz J, Ameigeiras P, Ramos-Munoz J J, Lopez-Soler J M 2013 Removing Redundant TCP Functionalities in Wired-cum-Wireless Networks with IEEE 802.11e HCCA Support *International Journal of Communication Systems*

[23] Caini C, Cornice P, Firrincieli R 2008 *IEEE Journal on Selected Areas in Communications* **26**(5) 820-7

[24] Caini C, Cruickshank H, Farrell S, et al 2011 Delay-and disruption-tolerant networking (DTN): an alternative solution for future satellite networking applications *Proceedings of the IEEE* **99**(11) 1980-97

[25] Yu Q, Sun X, Wang R, et al 2013 The effect of DTN custody transfer in deep-space communications *IEEE Wireless Communications* **20**(5) 169-76

[26] Celandroni N, Ferro E, Gotta A, et al 2014 On the applicability of reliable transport protocols in satellite delay tolerant and disruptive networks *International Journal of Satellite Communications and Networking* **32**(2) 141-61

[27] Omueti M, Trajkovic L 2007 TCP with adaptive delay and loss response for heterogeneous networks *Wireless Internet Conf.(WICON)*

[28] Omueti M, Trajković L 2007 OPNET Model of TCP with adaptive delay and loss response for broadband GEO satellite networks *Simon Fraser University*

[29] Fu C P, Liew S C 2003 TCP Veno: *IEEE Journal on Selected Areas in Communications* **21**(2) 216-28

Authors

Liang Zong, 1982, Jinshi, Hunan province, China.

Current position, grades: PhD candidate in the College of Information Science & Technology, Hainan University, China.
University studies: BSc of Sichuan University of Science and Engineering in 2006, Zigong, China, MSc of Ningbo University in 2010, Ningbo, China.
Scientific interest: maritime communications and computer networks.

Wencai Du, 1953, Xuzhou, Jiangsu province, China.

Current position, grades:
University studies: Master Degrees from ITC, The Netherlands and Hohai University, China respectively, the PhD from South Australia University, Australia and, Post-doctor Fellow in Israel Institute of Technology (IIT), Haifa, Israel.
Scientific interest: several aspects of Information Technology and Communication (ITC), including computer network and maritime communication.
Publications: 18 books and more than 80 scientific publications.

Yong Bai, 1970, Zhengzhou, Henan province, China.

Current position, grades: professor at College of Information Science & Technology, Hainan University since 2010.
University studies: MSc degree from Beijing University of Posts and Telecommunications (BUPT), China, in 1995, PhD degree from Rutgers-The State University of New Jersey in 2001.
Scientific interest: mobile communications, and maritime communications.