Enhancement of HSWA-TCP Congestion Avoidance Scheme for High Speed Satellite Communication Network

Riddhi P. Pandya[1] Ayesha S. Shaikh[2]
[1,2]Assistant Professor
[1,2]Department of Information Technology
[1,2]CSPIT, CHARUSAT, At. Post. Changa, Anand, Gujarat-388421, India

Abstract— The Transmission Control Protocol (TCP) is the mostly used protocol for data transmission over terrestrial-wire line networks. However, TCP's congestion control algorithm wasn't created to match the special characteristics (long delay and high Bit Error Rate) of satellite links, resulting in a drastically performance degradation. For high speed satellite communication networks to match its characteristics such as long delay and high Bit Error Rate several protocols are designed. SCPS-TP is one of the protocols designed for satellite networks but its throughput is low in high speed satellite environment. SCPS-TP congestion avoidance algorithm is modified to adapt the characteristics of high speed satellite communication network which is called the HSWA-TCP. HSWA-TCP is simulated and results show that performance of HSWA-TCP decays very sharply in high BER. To overcome these limitations Enhanced HSWA-TCP protocol is developed which depends not only on RTT and BER but also on bandwidth estimation. Enhanced HSWA-TCP continuously estimate bandwidth utilized by the sender, and based on that adjusts the cwnd rather than just at the time of packet loss. Simulation results show that, Enhanced HSWA-TCP throughput is improved and remain constant even in high BER and on average, throughput of Enhanced HSWA-TCP is 56% improved compare to HSWA-TCP.

Key words: HSWA-TCP, Satellite Communication, BER

I. INTRODUCTION

Transmission capacity for satellite networks is increasing from hundreds of Mbps to 1 Gbps along with development of technologies in satellite communication. High speed satellite communication networks are expanding and inspiring all research organizations to develop a technology which can cover the globe for the commercial purpose in future. And that is why it is important to improve the performance of the TCP to meet the needs of high speed satellite networks.

We know, traditional TCPs are made for terrestrial wire-line networks. Satellite communication networks are having characteristics of long delay and high Bit Error Rate (BER). From the view of TCP, the throughput is inversely proportional to the round trip time (RTT) of a connection, and is approximately proportional to the congestion window (cwnd), which represents the amount of unacknowledged data the sender can have in transit towards the receiver [1].

In Satellite networks, TCP throughput decreases [2] because the long propagation delays cause longer duration of the Slow Start phase during which the sender may not use the available bandwidth. The TCP protocol was initially designed to work in networks with low link error rates, i.e., all segment losses were mostly due to network congestions. As a result the sender decreases its transmission rate each time a segment loss is detected. This causes unnecessary throughput degradation if segment losses occur due to link errors, as it is likely in satellite networks.

Although TCP performance over satellite communication networks is limited due to satellite network characteristics, traffic control is still necessary for reliable data transport over satellite communication networks. There are many enhancements have been recommended to improve the TCP performance in space environment [3]: Link layer solutions, Performance enhancement proxy (PEP) solutions and end-to-end solutions. Link layer solutions mainly adopt the technologies of Automatic repeat request (ARQ) and Forward error correction (FEC) to lessen packet losses due to the high BER over satellite links [4].

PEP solutions can be further divided into TCP spoofing proxies and TCP split connection proxies [5]. TCP spoofing proxies solve the problem of slow start of TCP over long delay networks by locally acknowledging TCP segments [6]. To overcome the problems of long congestions control feedback delay and inability to differentiate between packet loss due to congestion or transmission errors, TCP Split connection have been presented to compensate for specific satellite link characteristics that cause the TCP performance degradation [5]. But the problem with TCP split connection is that by breaking the end-to-end connection, it is no longer reliable or secure.

HSWA-TCP is most suitable protocol so far for end-to-end solution for high the satellite network which solves the drawback of SCPS-TP [8] but yet it has disadvantage of low throughput in high BER. Enhanced HSWA-TCP is developed to overcome the problem of HSWA-TCP and maintain throughput in high BER for single as well as multiple connections.

Section II describes state of the art related work in the context of TCP performance over satellite communication networks. In Section III, we discuss throughput limited to problem analysis of HSWA-TCP protocol through its simulation results. Simulation Experimental setup and assumed parameters are stated in Section IV. Section V describes the developed protocol Enhanced HSWA-TCP and discuss its simulation results. Section VI concludes with recommendation and direction of future work.

II. LITERATURE REVIEW

High speed TCP [7] is an adoptive window algorithm proposed for operations in networks with a large BDP. The increment and decrement in congestion window, in response to an ACK reception or to a packet loss, respectively depend on the current value of window size. But Values of Parameters such as Low_P, High_p, Low_Window and High_Window are not suitably defined for a satellite environment. And HSTCP modified response function would only take effect with higher congestion window and with low congestion [7].

Scalable TCP is a simple change to the traditional TCP congestion control algorithm (RFC 2581) which dramatically improves the TCP performance in high speed wide area networks [13]. S-TCP is based on a Multiplicative Increase Multiplicative Decrease (MIMD) algorithm, according to which the congestion window is increased by a factor ‘α’ upon an ACK reception and decreased by a factor ‘β’ upon a packet loss event. But the parameters used in adjusting the CWND are not optimized for GEO satellite networks.

TCP-Peach [8] is composed of two new algorithms - Sudden Start and Rapid Recovery as well as two traditional algorithms, Congestion Avoidance and Fast Retransmit as shown in fig. 2.3. Sudden Start and Rapid Recovery replace the Slow Start and Fast Recovery algorithms, respectively. TCP Peach uses the unusual concept of dummy-segments to probe the availability of network resources. The dummy-segments do not carry any new information to the sender, and are therefore low-priority segments and will be dropped first in case of congestion. Due to the low-priority, the segments do not cause any decrease of data throughput of actual data. The dummy-segments have one or more of the unused flags set in the TCP-header, the responding ACKs have the same flags set. If the TCP receiver does not support dummy-segments, the TCP Peach sender stops sending dummy-segments and start to behave like TCP Reno. Disadvantage of TCP-Peach is it requires packet prioritization mechanism at every intermediate router along the data transmission path.

The information conveyed by ACK for data segments after rapid recovery in [8] is not fully utilized. Window Expanding Period introduced in [9] lasts for one RTT which is the period from the time the sender exiting/quitting the rapid recovery algorithm to receiving the ACK of the last data segment sent in the rapid recovery. TCP-Swift protocol develops two new algorithms, namely Speedy Start and Speedy Recovery, to improve the throughput performance in satellite communication networks [9]. In the field of satellite communication networks, the most successful working protocol so far is SCPS which is mainly developed by NASA and the U.S. DoD [7]. The SCPS suite includes four protocols, and one of these four protocols is SCPS-TP which is used to implement the transmission control protocol in satellite communication networks [10]. However there are still some limitations of SCPS-TP related with slow start phase and congestion control phase. HSWA-TCP, explained in detail in following section, is a congestion avoidance algorithm is based on dynamic Network transfer delay, where Network Transfer Delay is combination of Propagation delay (constant), transmission delay, and queuing delay. Problem of this protocol is that its throughput decays sharply in high BER due to retransmission of many lost packets which transfers the control to congestion control phase which is the same as standard TCP.

III. PROBLEM ANALYSIS OF HSWA-TCP

HSWA-TCP congestion avoidance algorithm is based on dynamic Network transfer. Main task offered by an author is to design an efficient traffic algorithm by considering variable transfer delay of every transmitted packet. Before describing algorithm, few variables are defined which are used in algorithm: Unit_RTT is the half of the minimum RTT measured during the past observation time. AvgRTT is the average of all RTTs measured before the current packet is sent. cwnd is the size of congestion window. cwnd_lim is the maximum size of congestion window. Increment is the adjusting size of congestion window. p is the Bit Error Rate (BER) in satellite link.TD is the current difference of measured AvgRTT and Unit_RTT. BaseTD is the minimum of all differences of measured AvgRTT and Unit_RTT Considering relationship between BaseTD and TD, cwnd adjusting scheme of HSWA-TCP has been designed by the author as follows:

$$Increment = \begin{cases} \min(S, cwnd\_lim - cwnd) & ; TD = BaseTD \\ S * \left(1 - \frac{TD - BaseTD}{BaseTD}\right) & ; BaseTD < TD < 2 * BaseTD \\ 0 & ; TD \geq BaseTD \end{cases} \quad (3.1)$$

Where S is the window adjusting rate of congestion window. From the simulation study, author has found that S depends on the Bit Error Rate of the network and suggested S = -A*log2 p, where p is the Bit Error Rate (BER) in a satellite link. And A is some constant. Through the simulation, author has found that value of 10 multiplied with log2 p gives the best performance so the value of A is fixed as 10 in the algorithm.

The congestion avoidance scheme shown above is divided into three sub-phases. In the case of TD = BaseTD, the Network link is in perfect phase and is called non-congestion sub-phase. In the non-congestion sub-phase, the increment of congestion window could be set as S if the difference of cwnd_lim and cwnd is greater than S. In the case of BaseTD< TD < 2*BaseTD, the network link is in potential congestion phase and is also called as pre-congestion sub-phase. In which, the increment of congestion window (cwnd) proportionally decreases with the increase of TD, which can be illustrated in figure below.

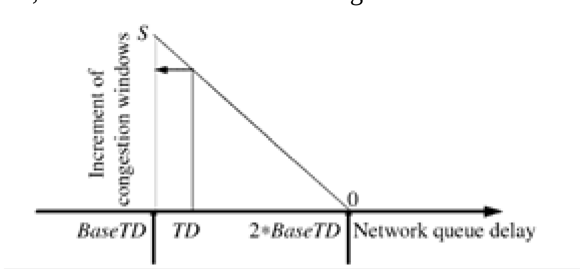

Fig. 1: Increment of cwnd in pre-congestion sub-phase [3]

In case of TD > = 2* BaseTD, the Network transfer delay is large. Although network congestion has not happened yet, it has high probability that network congestion level will arise. In this case the increment of congestion window is zero to avoid network congestion.

Simulation of HSWA-TCP has been done using the architecture shown in Fig. 4. We have also done simulation of various flavors of TCP as well as SCPS-TP (disadvantage of it HSWA-TCP overcome) to compare the results of HSWA-TCP with them. Fig. 2 and Fig. 3 show experimental result of simulation for single and multiple connections respectively.

In Fig. 1, Simulation result shows that HSWA-TCP’s congestion window adjusting scheme allows bandwidth utilization more aggressive when the current window size is closer to perfect point or BER is low. And less aggressive when current window size gets closer to saturation point or BER high.

From Simulation and Experimental results we have found some drawbacks of HSWA-TCP: one drawback is that its throughput decays sharply in high BER due to retransmission of many lost packets which transfers the control to congestion control phase which is the same as standard TCP. Another shortcoming is HSWA-TCP adjusts the cwnd only when congestion occurs (e.g. packet loss indicated by 3 duplicate ACKs or timeout expires). So the situation of congestion is awaited to adjust the cwnd but not prevented. To overcome this, we have developed Enhanced HSWA-TCP Algorithm.

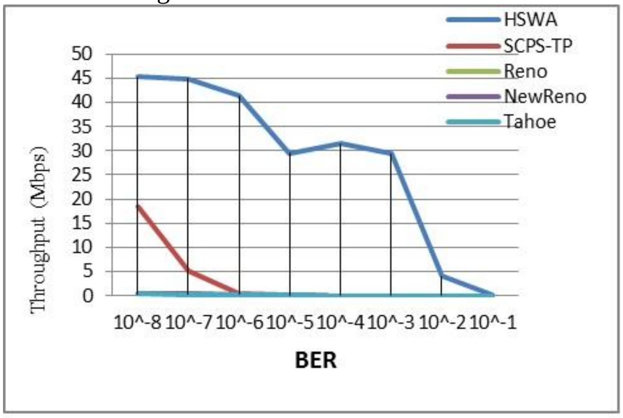

Fig. 2: Throughput vs. BER of different TCP flavors & HSWA-TCP

Since the high speed satellite communication link will undertake the backbone link in the future global communication networks, the case of multiply TCP connections simultaneously sharing the same satellite link will widely exist in high speed satellite communication networks. Therefore, it is important to analyze the throughput performance and bandwidth utilization with multiply TCP connections. Fig. 3 demonstrates throughput versus BER for multiple connections (3, 5, 10 and 15). It shows that as the number of connections reduces, throughput increases. For 3 connections, throughput remains almost equal for the BER 10-8 ~ 10-4. And for high BER 10-3 it starts deteriorating sharply. For the 5, 10 and 15 connections throughout drops highly compare to 3 connections.

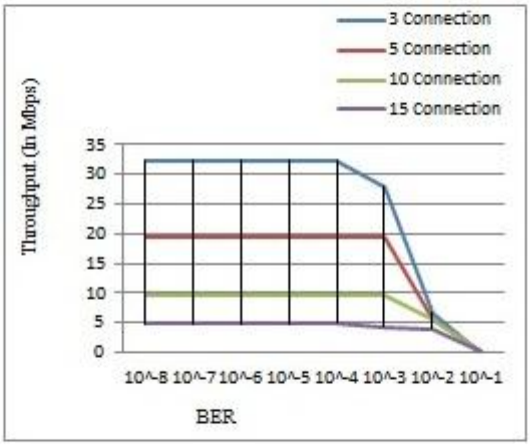

Fig.3. Average throughput of HSWA-TCP with multiply connections Vs. BER

IV. SIMULATION CONFIGURATION

In order to analyze the performance of the new algorithm, we build system architecture for the deployment of the proposed algorithm in the NS2 simulation platform and TCP Li n control algorithm on NS2, with similar simulation speeds and memory usages.nux Patch. TCP Linux patch is the patch that can run the linux TCP congestion control algorithm on NS2, with similar situation speeds and memory usages.

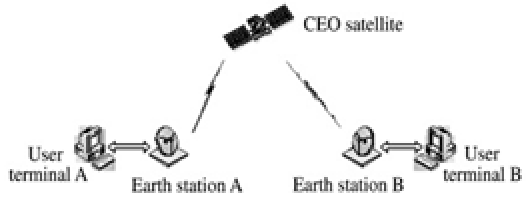

Fig. 4: Simulation Setup of Experiment [3]

The system architecture is illustrated in Fig.4. The satellite network provides an intermediate link in the end-to-end connection. We consider a simple bent-pipe satellite that relays packets received from the uplink to the downlink without demodulation or error checking. There are two gateway satellite earth stations at either end of the satellite link. The transmission control protocol is realized by incorporating different congestion avoidance algorithms in respective user terminals.

Parameter	**Value**
Number of User Terminals	2
Number of Earth Stations	2
GEO Repeater	1
Link Capacity between User Terminal and Earth Station	100 Mb
Type of Link between User Terminal and Earth Station	Wired
Link Capacity between GEO Repeater and Earth Station	100 Mb
Type of Link between GEO Repeater and Earth Station	Satellite Link
Round Trip Time (Between Earth Station & GEO)	500 msec
Round Trip Time (Between Terminal & Earth Station)	5 msec
Simulation Duration	50 sec
Channel Data Rate	100 Mbits/s
Queue limit	BDP

TABLE 1: Parameters and their values assumed in simulation

In our simulations, the File transfer protocol (FTP) has been used as the application data source. As shown in Fig. 4, a user terminal A connects with earth station A and sends data to the GEO satellite which is located at 36000km space, and then the GEO satellite relays the data to user terminal B through earth station B. Other parameters used in simulation are narrated in Table I. with assumed values.

V. ENHANCED HSWA-TCP ALGORITHM

Enhanced HSWA-TCP depends on BER, RTT, and Bandwidth Estimation. On the other hand HSWA – TCP depends only on RTT and BER to adjust its congestion window (cwnd). Another difference between these two is, HSWA-TCP adjust the cwnd only when the congestion occurs (indicated by packet loss as 3 duplicate ACKs or timeout expiration), whereas Enhanced HSWA-TCP continuously estimate bandwidth utilized by the sender, and based on that adjust the cwnd. How the Bandwidth is estimated is explained later section. Enhanced HSWA–TCP assumes packet loss or congestion when RTT is greater than AvgRTT. And depending on the condition state (true or false), it adjusts the congestion window. Fig. 5 demonstrates the full algorithm.

A. Bandwidth Estimation Technique Used

The Enhanced HSWA-TCP sender uses the ACKs to estimate the BWE. More precisely, the sender uses the following information: 1) the ACK arrival times and, 2) the increment of data delivered to the destination. Suppose an ACK is received at the source at time tk, notifying that dk bytes have been received at the receiver. Sample bandwidth used by the connection can be measured as $b_k = \frac{d_k}{t_k - t_{k-1}}$, where tk-1 is the time, when previous ACK was received. If $\Delta t_k = t_k - t_{k-1}$, then $b_k = \frac{d_k}{\Delta t_k}$.

Since congestion occurs whenever the low frequency input traffic rate exceeds the link capacity [15], a low–pass filter is employed to average sampled measurements and to obtain the low-frequency components of the available bandwidth.

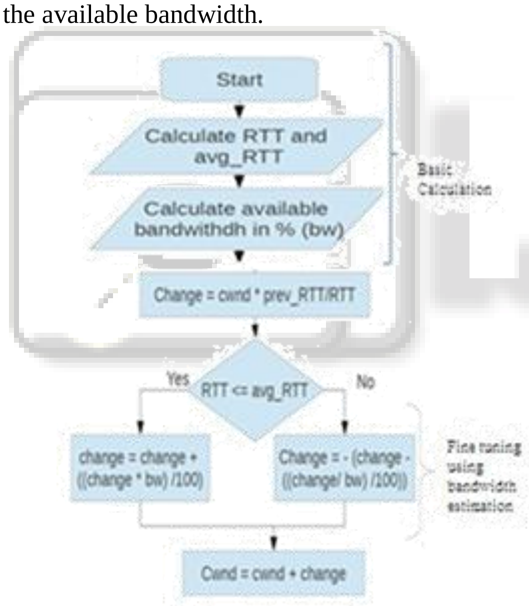

Fig.5. Algorithm of Enhanced HSWA-TCP

Let b_k be the bandwidth sample. The filter is then given by:

$$\hat{b}_k = \alpha_k * \hat{b}_{k-1} + (1 - \alpha_k)\left(\frac{b_k + b_{k-1}}{2}\right) \quad (5.1)$$

α_k is the time varying exponential filter coefficient at time t_k, Where $\alpha_k = \frac{2\tau - \Delta t_k}{2\tau + \Delta t_k}$ and $1/\tau$ is the filter cut off frequency. α_k Depends on t_k to properly reflect to the variable inter-arrival time

B. Experimental Result of Enhanced HSWA-TCP

Simulation has been done considering the same parameters as in Table no. 3.1.buffer size between sender and receiver is set equivalent to bandwidth delay product to assume only transmission errors. Fig. 6 shows Throughput of HSWA and Enhanced HSWA for single connection. In lowest BER, 10-8, which means almost no error, throughput of HSWA is maximum 96.98 Mbps. From 10-7 to 10-5, throughput of HSWA remains around 96 Mbps.

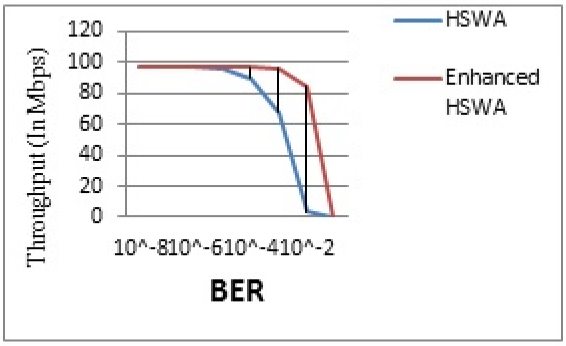

Fig.6. Throughput vs. BER of HSWA-TCP & Enhanced HSWA-TCP

But as BER starts increasing, performance of HSWA is disappointing. In 10-4, throughput is 89 Mbps. In 10-3 it reduces to 67 Mbps and when BER is highest (i.e. 10-2 and 10-1), throughput is lowest and reaches to almost 0. On the other end, throughput of Enhanced HSWA-TCP is around 96 Mbps throughout from 10-8 to 10-3 BER. In 10-2 BER throughput reduces around 10 Mbps and reaches to 85 Mbps. So we can see from graph that even in high BER conditions (i.e. 10-3 or 10-2), throughput of Enhanced HSWA-TCP does not deteriorate sharply and remains almost constant. And so the overall average throughput of Enhanced HSWA-TCP is improved around 56% compare to HSWA-TCP.

Since the high speed satellite communication link will undertake the backbone link in the future global communication networks, the case of multiply TCP connections simultaneously sharing the same satellite link will widely exist in high speed satellite communication networks. Therefore, it is important to analyze the throughput performance and bandwidth utilization with multiply TCP connections. For the 3 connections, throughput is almost same compare to HSWA-TCP for the BER 10-8 ~ 10-3. And for the high BER it drops only 5 Mbps compare to 20 Mbps in HSWA-TCP. For the 5 connections, it drops to 10 Mbps and remains almost constant throughout all BERs. For the 10 and 15 connections throughput is very low in range of 7~9 Mbps.

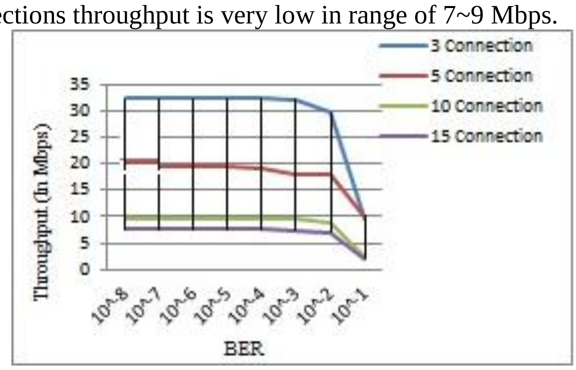

Fig. 7: Average throughput of HSWA-TCP with multiply connections Vs. BER

VI. CONCLUSION & FUTURE WORK

Parameter, bandwidth utilization by the sender apart from RTT and BER, is used to calculate the increment factor of congestion window. And in experimental result we shown that comparing to HSWA-TCP protocol, Enhanced HSWA-

TCP increases around 56% of its throughput and keeps it even in high BER. The number of more parameters available, more accurate result can be calculated. To improve the result such more parameters can be added in future and can improve the result for multiple connections as well. So that the ultimate goal of satellite communication, to maximum utilize the bandwidth available, is achieved and loads of data can be sent within a blinks of seconds.

REFERENCES

[1] W. Stevens, TCP/IP Illustrated. Reading, MA: Addison-Wesley, 1994, vol. 1.

[2] Partridge and T. J. Sheppard, “TCP/IP performance over satellite links,” IEEE Network Mag., pp. 44–49, Sept-Oct. 1997

[3] GE Xiaohu, LIU Jianhua, LIU Yingzhuang and DONG Yan, “HSWA-TCP: A New Traffic Control Algorithm for High-speed Satellite Communication Networks”, Chinese Journal of Electronics Vol.18, No.4, Oct. 2009

[4] S. Shambayati, C. R. Jones, D. Divsalar, ”Maximizing throughput for satellite communication in hybrid FEC/ARQ scheme using LDPC codes”, Proc. of IEEE MilCom 2005, Atlantic City, NJ, USA, Vol.5, pp.3289-3298, 2005.

[5] J. Border, M. Kojo, J. Grineret al., “Performance enhancing proxies intended to integrate ink-related degradations”, IETFRFC 3135, June 2001

[6] J. Ishac and M. Allman, “On the performance of TCP spoofing in satellite networks”, in Proc. MILCOM 2001, Vol.1, pp.700 - 704, 2001

[7] Robert C. Durst, Patrick D. Feighery, Eric J. Travis, “User's Manual for the SCPS Transport Protocol”, WN 97W0000018, Mar-1997.

[8] I.F. Akyildiz, “TCP-Peach: a new congestion control scheme for satellite IP networks", IEEE Transactions on Networking, Vol.9 No.3, pp.12-21, 2001.

[9] L.K. Fai, K.L. Yeung, “TCP-Swift: an end-host enhancement scheme for TCP over satellite IP networks”, Proc. of IEEE ISCC 2004, Alexandria, Egypt, Vol.1, No.1, pp.551-555, 2004

[10] M. Allman, V. Paxson and W. Stevens, “TCP Congestion Control”, RFC 2581, April 1999

[11] Hu Hsiao, Prof. Chienhuachen, “Performance evaluation of TCP Westwood and variant over wireless LANs”, July 2006