An Exhaustive Analysis of TCP-Variants for Satellite Networks

[1]Ketan R. Tandel, [2]Saurabh M. Patel, [3]Kirit R Bhatt

[1]Assistant Professor, [2]Assistant Professor, [3] Professor
[1,2,3] Department of Electronics & Communication Engineering,
[1,2,3] Sardar Vallabhbhai Patel Institute of Technology, Vasad, India.

Abstract : Current TCP protocols have lower throughput performance in satellite networks mainly because of long propagation delays and high link error rates. The survey of various Transmission Control Protocols in Satellite Network is important and necessary for smart transport system. This paper discusses the advantages/disadvantages of various Transmission Control Protocols for Satellite networks. It explores the motivation behind the designed, and traces the evolution of these Transmission Control Protocols.

***IndexTerms* - Congestion control, high bit error rates, long propagation delays, satellite networks, TCP protocols. Bandwidth estimation, explicit congestion notification, loss differentiation.**

I. INTRODUCTION

Today's packet switching technologies have eventually merged the traditional voice networks and data networks together into a converged and integrated multimedia network. The horizon of the converged integrated network is extending further to incorporate wired, wireless, and satellite technologies. All-Internet protocol (IP) wired and wireless hybrid network is becoming a reality. Transmission control protocol (TCP) has become the dominant communication protocol suite in today's multimedia applications. Nowadays, about 90% of the Internet traffics are carried by TCP. TCP needs to depart from its original wired network oriented design and evolve to meet the challenges introduced by the wireless portion of the network. IP [1] is a connectionless, best-effort-based variable length packet delivery network layer protocol that does not guarantee the reliable, timely and in-order delivery of packets between end stations. TCP [2] is a layer 4 transport protocol that uses the basic IP services to provide applications with an end-to-end connection-oriented packet transport mechanism that ensures the reliable and ordered delivery of data.

TCP was originally designed primarily for the wired networks. In the wired networks, random bit-error rate (BER) is negligible and congestion is the main cause of packet loss. TCP and many of its variants implement flow control and congestion control algorithms [3] based on the sliding window and additive increase multiplicative decrease (AIMD) [4] algorithms. A sliding window-based flow control mechanism allows the sender to advance the transmission window upon the reception of an acknowledgment (ACK) that indicates the last in-order packet has been received successfully by the receiver. When packet loss occurs at a congested link due to buffer overflow at the intermediate router, either the sender receives duplicate ACKs (DUPACK) or the sender's retransmission timeout (RTO) timer expires. These events activate the sender's congestion control mechanism by which the sender reduces the size of its transmission window, or congestion window ( ) in TCP terms, resulting in a lower transmission rate to relieve the link congestion.

The original TCP algorithm, TCP-Tahoe, has three transmission phases, namely slow-start (SS), congestion avoidance (CA), and fast retransmit. TCP-Reno [5] extends TCP-Tahoe to include a fast recovery phase in conjunction with the fast retransmit. TCP maintains two variables, the congestion window size ( ), which is initially set to be 1 maximum segment size (MSS), and SS threshold ( ). At the beginning of the TCP connection, the sender enters the SS phase, in which it increases the by 1 MSS for every ACK it receives. Therefore, in the SS phase, TCP sender's grows exponentially. When the reaches the , the TCP sender enters the CA phase. D-uring this phase, the sender increases the by for every ACK it receives, which is equivalent to an increment of by 1 MSS for every round-trip time (RTT). This additive increase leads to the linear growth of the transmission rate that helps the sender to slowly probe the available network bandwidth. The congestion window is reduced by 1/2 of the current value when the sender has received DUPACKs (is usually Three).

At this point, TCP infers that packets were lost due to link congestion; it sets the to be the same as and starts retransmission of the lost packet. This marks the beginning of the fast retransmit phase. During the fast retransmit phase, TCP-Reno invokes the fast recovery algorithm to speed up the recovery process, by which the sender treats the DUPACKs received during the fast retransmit phase as normal ACKs and artificially inflates the . This inflated portion of is later deducted at the end of the fast retransmit phase. The fast retransmit phase ends when the sender receives a normal ACK that acknowledges that the receiver has successfully received an ordered packet whose sequence number passes beyond the previous lost packet. Therefore, TCP decreases its multiplicatively in the presence of packet loss. The above TCP sender behaviour works fairly well in wired network where packet losses are almost always caused by link congestions, and packet losses due to bit errors are usually negligible or, if any, not exceeding one loss per transmission window.

However, in satellite networks where the connection having long propagation delay required more time to acknowledgement come back, so it takes larger time to utilize available bandwidth. This situation further becomes worst in case of lossy environment. Where, after loss detection cwnd is drastically decreased, so connection fails to reach available transmission rate. So, current standard TCP variants are failing in utilizing available network bandwidth in case of high Bandwidth Delay Products (BDP) links, and in network having wireless segment. So as a modification to current standard TCP congestion control algorithm many researchers proposed many different algorithms to address such problems. These proposed algorithms emphasizing on solving some problems while still having some trade-offs.

Current TCP protocols have lower throughput performance in satellite networks mainly due to effects of large propagation delay and high link error rates so, our aim is to study and analyse TCP enhancements for satellite networks. For that purpose we have selected three TCP-variants for solving issues of satellite networks. 1. TCP-Peach 2. TCP-Westwood 3. TCP- Jersey.

II. REVIEW OF TCP-VARIANTS

1. TCP-Peach [6]

TCP-Peach contains the following algorithms:

Sudden Start, Congestion Avoidance, Fast Retransmit, and Rapid Recovery, The Congestion Avoidance and Fast Retransmit algorithms may be those in TCP-Reno [3]. The new algorithms are based on the use of dummy segments.

Dummy Segments

Dummy segments are low-priority segments generated by the sender as a copy of the last transmitted data segment, i.e., they do not carry any new information to the receiver. The sender uses the dummy segments to probe the availability of network resources. If a router on the connection path is congested, then it discards the IP packets carrying dummy segments first. Consequently, the transmission of dummy segments does not cause a throughput decrease of actual data segments, i.e., the traditional segments. If the routers are not congested, then the dummy segments can reach the receiver. The sender sets one or more of the six unused bits in the TCP header to distinguish dummy segments from data segments. Therefore, the receiver can recognize the dummy segments and acknowledge them to the sender. The sender interprets the ACKs for dummy segments as the evidence that there are unused resources in the network and accordingly, can increase its transmission rate.

In Sudden Start, congestion window quickly reaches to the maximum available bandwidth compare to conventional TCP. So, it provides better utilization of available bandwidth in case of larger propagation delay link. Rapid Recovery replaces the classical fast recovery mechanism in an effort to improve throughput in the presence of high link error rate. In the rapid recovery phase, instead of trying to distinguish congestive loss from error loss, the sender uses the lower-priority dummy packets to interleave the data packets and inflates the sending window size upon receptions of ACKs for the dummy packets.

Drawbacks:

- Wastage of network resources since the delivery of dummy segments does not result in any gain in connection goodput.

(1) Implicitly assumed that more than half of the dummy segments are lost in transit for a congestive loss event

(2) All dummy segments can be successfully delivered to the destination for a non-congestive loss event

2. TCP-Westwood [7]

TCP Westwood (or TCPW for short), enhances the window control and backoff process. Namely, a TCPW sender monitors the acknowledgment stream it receives and from it estimates the data rate currently achieved by the connection. Whenever the sender perceives a packet loss (i.e., a timeout occurs or 3 DUPACKs are received), the sender uses the bandwidth estimate to properly set the congestion window (cwin) and the slow start threshold (ssthresh). By backing off to cwin and ssthresh values that are based on the estimated available bandwidth (rather than simply halving the current values as Reno does), TCP Westwood avoids reductions of cwin and ssthresh that can be excessive or insufficient. In this way TCP Westwood ensures both faster recovery and more effective congestion avoidance.

The key idea of TCP Westwood is to use the “bandwidth estimate” to directly control the congestion window and the slow start threshold. The bandwidth is estimated by monitoring the TCP ACKs. Namely, the source performs an end-to end estimate of the bandwidth available along a TCP connection by measuring and averaging the rate of returning ACKs. Bandwidth estimation (via ACK monitoring) has been used before to control the TCP window, but only indirectly, via the estimation of the bottleneck backlog [8,9]. After a congestion episode (i.e., the source receives three duplicate ACKs or a timeout) the source uses the measured bandwidth to properly set the congestion window and the slow start threshold, starting a procedure that we will call faster recovery. When an ACK is received by the source, it conveys the information that an amount of data corresponding to a specific transmitted packet was delivered to the destination. If the transmission process is not affected by losses, simply averaging the delivered data count over time yields a fair estimation of the bandwidth currently used by the source. When duplicate ACKs (DUPACKs), indicating an out - of sequence reception, reach the source, they should also count toward the bandwidth estimate, and a new estimate should be computed right after their reception.

Bandwidth Estimation

The general idea is to use the bandwidth estimate BWE to set the congestion window and the slow start threshold after a congestion episode. After the reception of n DUPACKs, TCPW sets both slow start threshold and congestion window equal to the ideal window BWE $*$ RTTmin. The standard Fast Retransmit/Fast Recovery (á la Reno) then follows. It should be noted that the value RTTmin is set to the smallest RTT sample observed over the duration of the connection. This setting allows the queue to be drained after a congestion episode. Also, note that after ssthresh has been set, the congestion window is set equal to the slow start threshold only if cwin > ssthresh. The rationale behind using BWE to set the slow start threshold is that TCP exploits the slow start phase to probe for available bandwidth; it thus seems natural to set ssthresh to the value we believe represented the available bandwidth at the time of congestion.

Drawbacks:

- TCP-Westwood behaves more aggressively than other Non-wireless oriented TCP schemes or some unfriendliness to TCP Reno.

3. TCP-Jersey [10]

TCP - Jersey consists of two key components, the timestamp-based available bandwidth estimation (TABE) algorithm and the congestion warning (CW) router configuration. TABE is a TCP-sender-side algorithm that continuously estimates the bandwidth available to the connection and guides the sender to adjust its transmission rate when the network becomes congested. TABE is immune to the ACK drops as well as ACK compression. Congestion Warning(CW)[12] is a configuration of network routers such that routers alert end stations by marking all packets when there is a sign of an incipient congestion. The marking of packets by the CW configured routers helps the sender of the TCP connection to effectively differentiate packet losses caused by network congestion from those caused by wireless link errors [11].

TCP-Jersey adopts Slow Start, Congestion Avoidance, and fast recovery from Reno but replaces Reno's fast retransmit with explicit retransmit and introduces the rate control procedure. The only difference between Reno's fast retransmit procedure and Jersey's explicit retransmit procedure is that unlike Reno's retransmit procedure that halves the current congestion window before starting the retransmission, explicit retransmit keeps the current cwnd. It leaves the adjustment of the congestion window to the rate control procedure. If an ACK is received without the CW mark, it proceeds as Reno, i.e., invoking SS or CA depending on whether or not the cwnd is below the ssthresh. If the received ACK or the nth DUPACK is marked with the CW bit, it calls the rate control procedure to adjust the window size and proceeds with SS or CA if it is an ACK or enters the explicit retransmit if it is the nth DUPACK. When the nth DUPACK is received without the CW mark, TCP-Jersey renders the packet drop is caused by a random error, and therefore it enters the explicit retransmit without adjusting the window size.

Drawbacks:

- It fails to handle burst losses.

III. Conclusion

As per the analysis it can be concluded that TCP-Peach has certain limitations of using dummy segments in satellite networks. Due to which its performance level deteriorates consuming more bandwidth by dummy segments. In case of TCP-Westwood the problem of bandwidth consumption can be rectified for satellite networks. But TCP-Westwood behaves more aggressively compared to other conventional TCP-variants in heterogeneous networks (unfriendliness to TCP Reno). The same issue remains with TCP jersey. But in case of bandwidth estimation TCP-Jersey performs far better.

References:

[1] J. Postel, "Internet Protocol," IETF, RFC 791, 1981.

[2] "Transmission Control Protocol," IETF, RFC 793, 1981.

[3] V. Jacobson, "Congestion avoidance and control," in Proc. ACM SIGCOMM,Aug. 1988, pp. 314–329.

[4] D. Chiu and R. Jain, "Analysis of the increase/decrease algorithms for congestion avoidance in computer networks," J. Comput. Networks, vol. 17, no. 1, pp. 1–14, June 1989.

[5] W. Stevens, "TCP slow start, congestion avoidance, fast retransmit and fast recovery algorithms," IETF, RFC 2001, 1997.

[6] I. F. Akyildiz, G. Morabito, and S. Palazzo, "TCP-Peach: A new congestion control scheme for satellite IP networks," IEEE/ACM Trans. Networking, vol. 9, pp. 307–321, June 2001.

[7] C. Casetti, M. Gerla, S. Mascolo, M. Y. Sanadidi, and R. Wang, "TCP Westwood: Bandwidth estimation for enhanced transport over wireless links," ACM Mobicom, pp. 287–297, July 2001.

[8] B.S. Davies and L.L. Peterson, Computer Networks: A Systems Approach, 2nd ed. (Morgan Kaufman, 1999).

[9] S. Keshav, A control-theoretic approach to flow control, in: Proceedings of ACM SIGCOMM 1991 (September 1991).

[10] Kai Xu, Ye Tian, and Nirwan Ansari," TCP-Jersey for Wireless IP Communications", IEEE Journal on Selected Areas in Communications, Vol. 22, No. 4, May 2004.

[11] R. C. Durst, G. J. Miller, E. J. Travis, TCP extension for space communication, in Proc.. ACM Mobicom, NOV 1996.

[12] H. Balakrishnan and R. H. Katz, "Explicit loss notification and wireless web performance," in Proc. IEEE GLOBECOM Internet Mini-Conf., Sydney, Australia, Nov. 1998.