ISSN: 2321-7782 (Online)
Impact Factor: 6.047

Volume 4, Issue 6, June 2016

International Journal of Advance Research in Computer Science and Management Studies

Research Article / Survey Paper / Case Study

Available online at: www.ijarcsms.com

Survey on Different Types of TCP Protocols for High Speed Networks

Dr. Syed Shabbeer Ahmad
Professor in CSE,
Muffakham Jah College of Engineering & Technology
Hyderabad – India

Abstract: ***New challenges arise with the presence of various types of physical links, such as wireless networks, high speed and satellite in today's ever-changing network. It is clear that the TCP throughput deteriorates in high-speed networks with large bandwidth-delay product, and new congestion control algorithms have been proposed to address such deterioration. Traditional TCP protocols treat all packet loss as a sign of congestion. Their inability to recognize non-congestion related packet loss has significant effects on the communication efficiency. The proposed protocols such as TCP Adaptive Westwood, Scalable TCP, HS-TCP, BIC-TCP, FAST-TCP and H-TCP all have some improvement in functionality over the traditional TCP protocols. This survey gives a summarization of all the protocols for high speed networks.***

Keywords: TCP protocols; high speed networks; TCP congestion.

I. INTRODUCTION

The Transmission Control Protocol (TCP) congestion control algorithm is successful in making the functioning of internet efficiently. However, it shows very poor performance over the networks with high bandwidth - delay product paths. The problem starts from the fact that the standard AIMD (Additive increase and multiplicative decrease) congestion control algorithm increases the congestion window too slowly. This would lead to probably long file transfer times. With the present algorithms, latencies of only a few tens of milliseconds are quire sufficient to create bandwidth-delay products that yield poor throughput performance [1]. Therefore innovative schemes were designed to provide a solution to certain issues. TCP designed mainly for wired networks is a connection oriented transport protocol that provides reliable data communications. The reason for the performance degradation is that the congestion control mechanism in TCP cannot distinguish between the packet loss caused by wireless link error and that caused by network congestion, thus, reacting to the loss by reducing its congestion window (cwnd). Therefore, these in appropriate reductions of the cwnd lead to unnecessary throughput degradation for TCP applications [2]. A solution to the problem has been given by many authors. The main aim is to increase the rate at which cwnd is increased and thereby shortens the congestion epoch duration and also it should be suitable to standard TCP paths with low bandwidth – delay product (BDP). The previous research on these lines includes the HS-TCP proposal [8], the scalable TCP [7] and the FAST – TCP [11] and many more recent proposals include BIC-TCP [8] and H-TCP [10]. The rest of the document is organized as follows. Section 2 offers an overview of the basic design of TCP protocol and its mechanisms. Section 3 discusses various protocols for high speed networks provided in the previous research papers. Section 4 describes the experimental results from the related work and Section 5 concludes the review of the study of various TCP protocols for high speed networks.

II. BASIC DESIGN OF TCP PROTOCOL

TCP is a transport layer protocol that provides a reliable and in-order delivery of data between two hosts. It is a defensive protocol highly sensitive to network congestion.TCP issues an acknowledgement packet (ACK) as a response to a successfully delivered packet to ensure a reliable communication. The standard TCP congestion control mechanism is based upon a sliding

window, which defines the number of packets injected in the network. The congestion window value is updated after the reception of an ACK and upon the detection of a packet loss. The reception of an ACK is interpreted by TCP as a signal of available bandwidth, so that the congestion window value, W, may be increased. Theincrease is fast at the beginning of a new connection (the socalled slow-start phase), where the TCP sender enlarges W by one segment for each received ACK. In this way, the congestion window grows exponentially on a Round Trip Time (RTT) basis. Once a given threshold value is crossed, the connection enters the congestion avoidance phase, where the TCP sender gently increases its transmission rate in a linear fashion over RTT (the congestion window is increased by 1 W upon an ACK reception). The decrease phase is triggered by the detection of a packet loss. A packet loss can be detected either by a timeout expiration or by the reception of 3 duplicate ACKs (correspondently, the congestion window is halved). Since the timeout granularity is fairly large, the first case is interpreted by TCP as a signal of severe network congestion so that it reduces its transmission rate to the minimum. In the other case, since some packets have been correctly received after the lost one(s), the congestion is assumed to be a transient phenomenon It is easy to verify that this mechanism is pretty inefficient in the presence of large W values (or,equivalently, large BDP values). Indeed, let us consider an error occurring in the presence of a congestion window value of W. Since W grows approximately by 1 for RTT, the time it takes to go back from W/2 to W equals Trec = (W·RTT) / 2 . This will be called as the time to recover from a loss. The fact that Trec depends linearly on W suggests that the TCP's AIMD mechanism does not scale well with BDP. This is one of the main reasons that have motivated the study of congestion control mechanisms able to perform well in a large BDP scenario. The packet loss is detected by (1) a timeout and (2) duplicate acknowledgement (ACK). A timeout occurs when the TCP sender does not receive any acknowledgement from the receiver even after a prescribed time. When timeout occurs, TCP treats the situation as network congestion and performs slow start. In the second case, when the TCP sender receives duplicate ACK, it identifies receiver received out-of order packets. TCP enters fast retransmit and fast recovery algorithm.

III. VARIANTS OF TCP

A. TCP Adaptive Westwood

TCP Adaptive Westwood (TCP-AW) is a combination of the best features from both TCP Westwood ABSE (TCPABSE) [4] and TCP Adaptive Reno (TCP-AReno) [5]. TCPABSE's best feature is its eligible rate estimation (ERE) mechanism, which helps predict imminent congestion. Essentially, the congestion window is adjusted according to the network congestion rate, and is not based on the traditional delay model. TCP-AReno's best feature is its use of packet loss interval time to adjust the congestion window, instead of using bandwidth estimation. This approach improves RTT-fairness even when accurate bandwidth estimation is not available. For the high-speed scenario, TCP-AW[6] remarkably improved the aggregate throughput of the control group. This is due to the delay-based nature of the protocol. The protocol feels the presence of incoming TCP standard Reno flows, and then reacts accordingly to grab the extra bandwidth whenever the bottleneck is less loaded. TCP Adaptive Westwood shows good throughput in High Speed networks and performs safely. TCP-AW obtains a substantial improvement in coexistence due to its embedded loss discriminator component. *B. HS-TCP* HS-TCP was introduced to achieve high throughput in high bandwidth-delay product links without requiring unrealistically low packet loss rates. The HS-TCP modifies the standard TCP's Additive increase and multiplicative decrease (AIMD) algorithm to improve its loss discovery time. Such alteration would only be effective when higher congestion windows are encountered. This implies that if the congestion window is smaller than a given threshold, it makes use of the Standard (TCP Tahoe) AIMD algorithm, otherwise the High Speed algorithm is used [3]. The Standard AIMD algorithm upon receipt of ACK and in the event of congestion, the window is respectively given as: $w = w + 1/w$ $w = 0.5*w$

Parameters a and b for the increase and the decrease of the AIMD algorithm are fixed at 1 and 0.5 respectively. The modified HS-TCP algorithm is as follows: upon the receipt of acknowledgement = $w + a(w)/ w$ And when congestion is encountered, the window (w) is, $w = w - b(w)w$.

The increase and decrease parameters thus vary depending on the current value of the congestion window. *C. Scalable TCP* STCP seeks to improve the loss recovery time of Standard TCP; this idea mirrors that of the HS-TCP [3]. For standard TCP and HS-TCP connection, the packet loss recovery times increase (or reduce) in a proportion as the connection's window size and round trip time (RTT) does. In a Scalable TCP connection, packet loss recovery times are proportional to connection's RTT only. For the STCP [7], the slow start phase of the standard TCP algorithm is not modified, but its congestion avoidance phase is modifies thus: for every acknowledgement received in a RTT, the congestion window (cwnd) is cwnd = cwnd + 1/cwnd (for Std TCP) cwnd = cwnd + 0.01 (for STCP) -------(1) and when congestion is encountered in a given RTT, cwnd = cwnd - cwnd * 0.5 (for Std TCP) cwnd = cwnd – cwnd * 0.125 (for STCP) ----(2) Similarly the evolution of STCP's cwnd is similar to that of the HS-TCP; its threshold window size and the modified algorithm in equation (1) and (2) are used only when the size of the congestion window exceeds threshold window size. Thevalues of 0.01and 0.125 are suggested for the increase and the decrease parameters. STCP default value for the threshold window size is given as 16 segments [7]. *D. BIC – TCP* BIC-TCP [8] employs a binary search algorithm to update its congestion window.Briefly a variable w1 is maintained which holds a value halfway between the values of cwnd just before and Just after the last loss event. The window update rule seeks to rapidly increase its window beyond a specified distance Smax from w1, and update cwnd more slowly when its value is close tow1. Multiplicative backoff of cwnd is used on detecting a packet loss with a backoff factor β of 0.8. It also implements an algorithm whereby upon low utilization detection, it increases its window more aggressively. This is controlled with two factors namely, low utility and utility check. In order to maintain backwards compatibility, it uses the standard TCP update parameters when cwnd is below the threshold. *E. FAST – TCP* FAST – TCP [9] is a delay based algorithm. It also includes rate packing. Rate pacing is a functional change and is thus it can be viewed as a part of congestion control algorithm. The FAST TCP flows typically converge quickly initially, flows may later diverge again to create significant and sustained unfairness. The main drawback of this is where the threshold is somewhat higher, owing to the standing queue created by the delay-based congestion control action used here. *F. H – TCP* H-TCP [10] uses the elapsed time Δ since the last congestion event, rather than cwnd, to indicate path bandwidth-delay product and the AIMD increase parameter is varied as a function of Δ. The AIMD increase parameter is also scaled with path round – trip time to mitigate unfairness between competing flows with different round-trip times. The AIMD decrease factor is adjusted to improve link utilization based on an estimate of the queue provisioning on a path.

IV. Experimental Results from the Related Work

The performance analysis relevant to the present study is derived from many papers. In [7], Kelly presents an experimental comparison of the aggregate throughput performance of scalable-TCP and standard TCP. In [11], aggregate throughput measurements are presented for FASTTCP and TCP-Reno. In all of these studies, measurements focus on aggregate throughput i.e., link utilization. Here efficiency as a function of queue size is not considered, nor fairness, friendliness, responsiveness and convergence times. In [9], throughput and cwnd time histories of FAST-TCP, HS-TCP, Scalable-TCP, and TCP-Reno are presented for a lab scale experiment test bed. Aggregate throughput, throughput Fairness and a number of other measures are presented. However, results are confined solely to an 800-Mb/s bottleneck link with a 2000-packet buffer. The impact of link rate, RTT, queue size, and level of Web traffic on fairness and responsiveness are not considered nor is the impact of queue size on efficiency. In [8], NS simulation results are presented comparing the performance of HS-TCP, Scalable-TCP, BICTCP, and standard TCP.

V. Conclusion

This article concludes by presenting a comprehensive survey of current research on running TCP in high speed networks from various experimental results evaluating the performance of Scalable-TCP, HS-TCP, BIC-TCP, FAST TCP and H-TCP. All the protocols studied are all successful in improving the link utilization in a relatively static environment with long-lived flows. And many of them showed poor responsiveness to changing network conditions. Though there are various schemas and mechanisms proposed, there is no single mechanism that can overcome the unreliable nature of network in a reliable way. Each

and every mechanism has its own advantages and disadvantages. In short, any mechanism will be effective based on the factors that are to be taken into consideration. To conclude this area is not completely explored to its maximum and still lot more research can be done towards establishing a basis for the development of new protocols.

References

1. M.Allman, "TCP congestion control with appropriate byte counting', IETF RFC 3465, Feb 2003.
2. Lakshman, T. V. and Madhow, U.: The performance of TCP/IP for networks with high bandwidth- delay products and random loss. IEEE/ACM Transactions on Networking, Vol. 5, 336–350, 1997.
3. Sally Floyd, RFC 3649, "Highspeed TCP for large congestion window", 2003.
4. R. Wang, M. Valla, M. Sanadidi, and M.Gerla, "Using adaptive rate estimation to provide enhanced and robust transport over heterogeneous networks," IEEE ICNP, Nov 2002.
5. Cesar Marcondes, Jerrid Matthews, Robert Chen, " A cross comparison of advanced TCP protocols in High speed and satellite environments",IEEE, pp 179-185, 2008.
6. T. Kelly, "Scalable TCP: Improving Performances in High speed Wide Area networks," Computer Commn., Rev., vol.33, no.2, pp. 83-91, April 2003.
7. C.Jin,D.X.Wei,S.H.Low,G.Buhrmaster, J. Bunn, D. H. Choe, R. L. A. Cottrell, J. C. Doyle, W.Feng, O. Martin, H. Newmann, F. Paganini, S. Ravot and S. Singh, "FAST TCP : From theory to experiments", IEEE Network, vol. 19, no. 1, pp.4-11, 2005.
8. L.Xu, K.Harfoush and I.Rhee, "Binary increase obstacle control for fast long-distance networks", in Proc. IEEE INFOCOM, HongKong, 2004.
9. D.J.Leith and R.N.Shorten, "H-TCP protocol for high speed long distance networks,"presented at thend Workshop Protocols Fast Long Distance Networks, Argonne, Canada, 2004.

