MX-TCP and HS-TCP as Possible Options to Overcome TCP Limitations in Multi-Hop Ad-Hoc Networks

Article · December 2015

DOI: 10.17148/IJARCCE.2015.412118

2 authors:

Aniket Deshpande
Broadcom Inc.

Ashok Kaushal
Centre for Development of Advanced Computing

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 12, December 2015

MX-TCP and HS-TCP as Possible Options to Overcome TCP Limitations in Multi-Hop Ad-Hoc Networks

Aniket Deshpande[1], Dr. Ashok Kaushal[2]

Research Scholar, Faculty of Engineering and Technology, Mewar University, Chittorgarh, Rajasthan, India [1]

Research Guide, Faculty of Engineering and Technology, Mewar University, Chittorgarh, Rajasthan, India[2]

Abstract: The Transmission Control Protocol is the most widely-used protocol on the internet and the rapidly growing wireless networks. However, it raises a number of issues within an ad-hoc network. It has to deal with new tough challenges such as disconnection and route failures due to mobility congestion and channel loss. In order to adapt TCP to this demanding paradigm, some improvements have been made. These challenges led to the development of High Speed Transmission Control Protocol (HS-TCP) and the Maximum Transmission Control Protocol (MX-TCP).

Keywords: Ad-Hoc Networks, HS-TCP, MX-TCP, Multi-Hop Wireless Networks.

I. INTRODUCTION

Ad-hoc networks are increasingly being deployed throughout the world. These networks are complex communication systems with wireless nodes that can be freely self-organized into arbitrary network topologies. Ad-hoc network uses the Transmission Control Protocol for the transfer of files between the nodes. An ad-hoc network is a special type of wireless network involving a pool of nodes that have the capability to communicating with each other without physical cabling. The interconnections between nodes can change on an arbitrary and continual basis. Nodes within each other's radio range interconnect via wireless links [3]. TCP has been widely used as a dependable data transfer protocol in computer networks. However, fair and effective allocation of resources of a network such as bandwidth among a collection of competing users cause traffic congestion. Therefore, TCP faces challenges that HS-TCP and MX-TCP attempts to address in an ad-hoc network.

II. KEY TCP PERFORMANCE ISSUESIDENTIFIED IN MULTI-HOP WIRELESS NETWORKS

This section discusses the major problems that arise with TCP on Multi-Hop Wireless Ad-Hoc Networks.

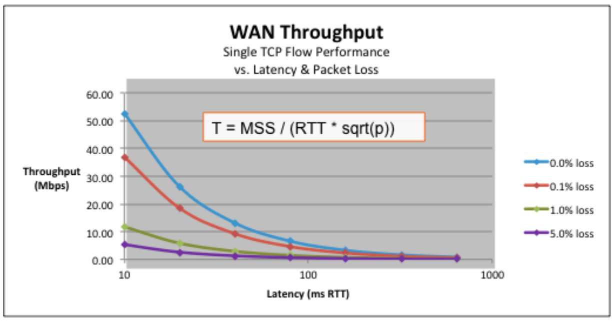


Fig.1 –A line graph showing default behaviour of TCP in terms of variation in throughput when subject to packet losses and latency.

A. Disconnection / Routing Failures

Route failures are a major problem in ad-hoc networks. The node mobility causes frequent topology changes that lead to route failures in mobile ad-hoc networks. Moreover, the link failures due to the conflict on the wireless network also lead to route failures in ad-hoc networks. In the event of a route failure, packets that are dropped from intermediate nodes along the route. This large amount of packet loss causes a series of time-outs at the TCP sender. In addition, since the route re-establishment after route failures is dependent on the underlying routing protocol, TCP lacks an indication of the route re-establishment duration. Moreover, after the route is restored, its initial sending rate is reduced.

B. Hidden and Exposed Terminals

Sharing of the bandwidth among ad-hoc connection poses a challenge to medium access control (MAC). It relies on CSMA/CA to determine the available channel such as the IEEE 802.11 distributed coordination function [3].

C. Congestion

TCP experiences network congestion through its attempt to utilize the network bandwidth. Due to factors such as unpredictable variable MAC delay and route change, the relationship between transfer congestion window size and the data rate tolerable for a route is not maintained in ad-hoc networks. The congestion window size for the old route may be too large for the newly established route, resulting in network congestion. If the sender continues transmitting at the full rate computed by the old congestion window size, congestion results in increased link contention and buffer overflow hence reducing TCP performance [3]. The capacity of wireless ad-hoc networks reduces as traffic or competition amongst the nodes arises.

Congestion control is a major challenge in ad-hoc networks. The standard TCP congestion control mechanism is unable to handle the special properties of a

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 12, December 2015

shared ad-hoc channel effectively. In particular, changes of the network topology and the shared nature of the network channel pose major challenges.

D. Intra-flow and inter-flow contention: packets compete for airtime
In shared ad-hoc networks, the throughput of each single node is limited by both the raw channel capacity and the transmissions from other nodes. Inter-flow contention is the contention experienced by a node due to transmissions from other nodes around it. Intra-flow contention is the contention for the shared network channel that a node experience as a result of forward data transmissions and the reverse acknowledgments of the same flow [1]. Thus, each ad-hoc flow encounters contentions from other flows that pass through the neighborhood and from the self-transmissions.

E. Channel Errors
Bit errors can corrupt packets in transmission hence leading TCP data packets or acknowledgments loss. If the TCP sender cannot receive the acknowledgments within the retransmission timeout, it immediately reduces its congestion window to a single packet and retransmits the lost packet [1]. Therefore, intermittent channel errors may cause the sender's congestion window size to remain small, leading to low throughput. If retransmission timeout expires or sender receives three duplicate acknowledgments and network state does not detect as congestion at the receiver's end, the sender assumes that packet loss is due to channel error. Since packet loss is random, the sender retransmits the lost packet.

III. TRADITIONAL BEHAVIOUR OF TCP AND ITS VARIANTS

A. TCP
TCP allows nodes to communicate over networks and enhances sharing of network bandwidth across network connections in a fair manner. It does so by allowing each TCP sender to adjust its transmission window, which signifies the maximum rate of data transfer through the network at any given time.

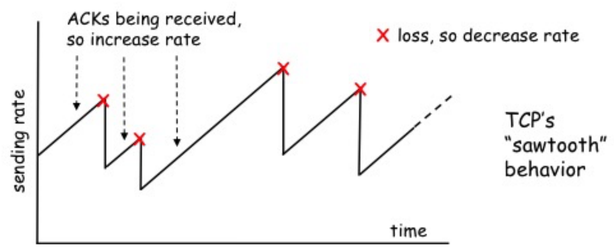


Fig.2 - A line graph showing the default 'Saw-Tooth" behaviour of the TCP Protocol with increasing delay in the receipt of acknowledgement followed by eventual loss of acknowledgment.

Window-adjustment algorithm causes TCP performance to degrade beyond 100M bit/sec [2]. In order to avoid congestion, ordinary TCP improves its transfer window by a packet every round-trip and when if congestion is detected, it reduces the transfer window in half [3]. For a high-latency and high-bandwidth connection, thus, it takes several round trips to recover from congestion. In an ad-hoc network, with lots of connections coming and going, normal TCP is too slow to track all of the activity.

B. HS-TCP
High-speed TCP (HSTCP) protocol takes advantage of high capacity bandwidth of network connections.It can support the large amount of congestion window as opposed to TCP [1]. It changes how the window is opened and closed on congestion occurrence as a function of the total size of the window. If the window is small, HS-TCP behaves exactly like an ordinary TCP. However, if the window is large, HS-TCP increases the window by a larger amount choosing the amounts based on the precise value of the window in operation. These changes eliminate the sluggishness of ordinary TCP [2]. HS-TCP performs well and enables full utilization of multi-gigabit, high-delay links. It is advancement to the mechanisms of TCP's current congestion control for use with TCP connections that have large congestion windows. The current ordinary TCP's congestion control mechanisms constrain the congestion windows to be achieved by TCP in realistic environments.

The larger increase and smaller decrease in window size makes HS-TCP recover faster than ordinary TCP mostly based on magnitude. This approach permits full utilization of high-speed ad-hoc links, and it does not lose or compromise any of the common and vital benefits of TCP. It works properly even when HS-TCP connections share network links with ordinary TCP links. A single HS-TCP connection has roughly the same congestion control performance as several ordinary TCP connections, while virtual connections increasing with the size of the window [2]. Some non-TCP protocols are not TCP-friendly and can hinder ordinary TCP connections. In contrast, an ordinary TCP connection experiences similar congestion control behaviour as if there were many other ordinary TCP connections using the network when it shares network with a HS-TCP connection.

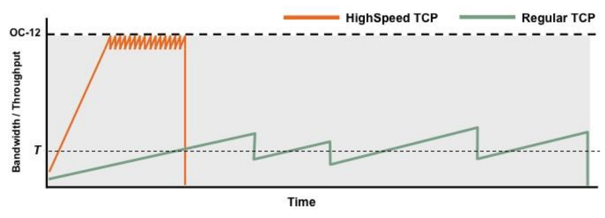


Fig. 3A line graph showing default behaviour of HS-TCP Protocol ramp-up and backing-off upon detecting congestion (especially packet losses) when compared to Regular TCP

C. MX-TCP
Maximum Speed TCP (MX-TCP) is a high acceleration type that allows devices to achieve maximum throughput for challenging environments with heavy congestion. It optimizes high-loss links where regular TCP would cause underutilization. MX-TCP removes the TCP congestion control algorithm from the inner connections thus allowing

the link to be saturated faster and eliminates chances of underutilizing the link. These environments have heavy network connections where the network is congested, but the delay is affection throughput. MX-TCP most suited environments include ad-hoc connections that often have high packet loss. MX-TCP is simply TCP without the congestion control challenges. MX-TCP simply supports traffic through the network as fast as it can go [3].

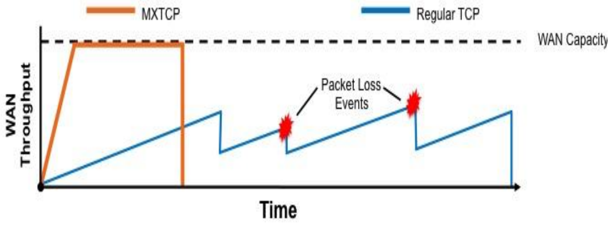


Fig. 4 A line graph showing default behaviour of MX-TCP Protocol which doesn't back-off upon detecting congestion but merely prioritises lost packets to be sent again.

MX-TCP claims to offer a better alternative to the default TCP behaviour. It effectively allows one to set a bandwidth, and send data onto the network at that defined rate. There is one potential problem to this method. Since MX-TCP requires disabling TCP's congestion control, the MX-TCP is not fair to other traffic competing for bandwidth, so any other flows that are still using normal TCP will now back off even further. However, if the whole of the transmission channel is designated for maximum traffic, or if it is allocation of a specific percentage of the bandwidth to maximum traffic, MX-TCP will provide for just that. With MX-TCP, any packets lost due to congestion, under-buffered router interfaces, or other impairments will be immediately retransmitted without the Steelheads backing off. The result is that the channel fills up to the level dialled in during configuration of MX-TCP.

IV. COMPARING THE DIFFERENCES INTHE DEFAULT BEHAVIOUR HS-TCP AND MX-TCP PROTOCOLS

TCP interacts with routers in the subnet and reacts to implicit congestion notification (packet drop) by reducing the TCP sender's congestion window.TCP increases congestion window using slow start or congestion avoidance. Lossy networks thus cause traditional TCP transfers to ramp up slowly and ramp back down at the first sign of packet loss. This causes the jagged sawtooth pattern in blue.

In comparison, HS-TCP achieves full utilization of network bandwidth without compromising or losing any of the essential characteristics and benefits of TCP. These benefits include safe congestion control, even when HS-TCP connections share network links with normal TCP connections. Familiar TCP performance features have been conserved [2]. For example, there is no need to determine obtainable bandwidth in advance because HS-TCP adjusts transmission rate automatically. Similarly, High-Speed TCP avoids slow start, but will still back off in congestion.

In contrast, MX-TCP allows users to take advantage of 100 per cent of a prescribed amount of bandwidth connectivity between any two locations. Whereas HS-TCP will back down in speed as a result of significant packet loss or congestion, MXTCP is designed to use a set amount of bandwidth regardless of congestion or packet loss [3]. Administrators can easily set the bandwidth limit for MX-TCP, allowing the functionality without the need of total bandwidth on the connection.

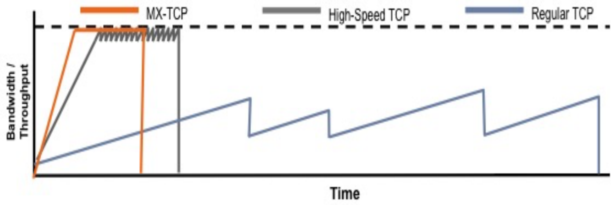


Fig. 5 A line graph showing comparative behaviour (Throughput / Bandwidth Utilisation) of Regular TCP, HS-TCP and MX-TCP when subjected to similar volume of data transfer, rate of packet-loss, latency and available bandwidth.

V. SUGGESTED TECHNIQUE TO OVERCOME THE TCP PERFORMANCE CHALLENGES USING MX-TCP AND HS-TCP VARIANTS

High-speed TCP (HSTCP) and MXTCP help solve some of the TCP challenges. Both protocols address the problem of congestion and disconnection. When the congestion window becomes large, it increases the window by a larger amount [1].

As discussed above and shown in the below figure 5, HS-TCP ramps up faster and backs-off more slowly when congestion occurs hence enabling greater utilization of a large link. Therefore, use of HS-TCP in an ad-hoc with many connections minimizes effects of traffic congestion especially when a single TCP connection is being used for bulk data transfer.HS-TCP works well when this connection must share the bandwidth with other connections, and there is a "clean" circuit, i.e., no packet loss. However, as we observe below, when the window becomes small together with increased rate of packet losses, HS-TCP behaves like an ordinary TCP.

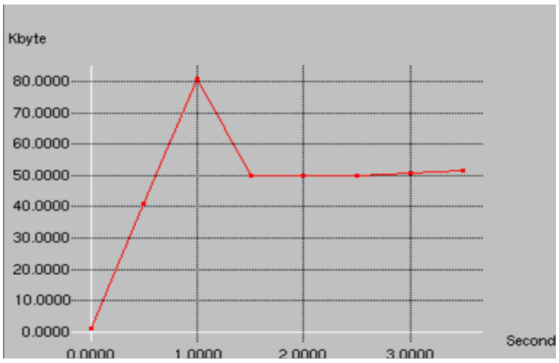


Fig. 6 A line graph showing default behaviour of HS-TCP Protocol ramp-up and back-off upon detecting congestion

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 12, December 2015

On the other hand, MX-TCP allows the administrator to set a throughput limit, and uses 100% of that limit hence reducing the problem of disconnection [2]. In the event of packet loss, MX-TCP does not back off but prioritizes the lost packets to be sent again.

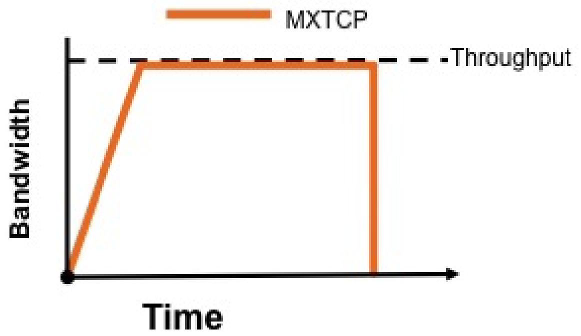
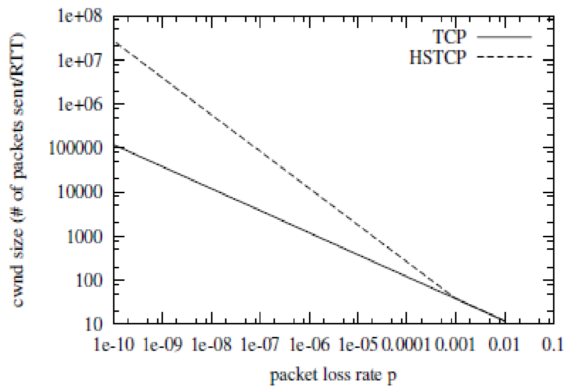


Fig. 7 A line graph showing default behaviour of MX-TCP Protocol sustaining higher throughput at a continuous rate irrespective of packet losses or link latency and solely dependent on the bandwidth preconfigured by the administrator till the completion of session.

VI. CONCLUSION

HSTCP has a greater throughput than TCP in connections with high bandwidths, and attains the same amount of transfer as TCP. HSTCP changes the standard TCP transfer function to acquire faster the available bandwidth [1]. HSTCP is well adapted to very high transfer speeds and performs better than TCP in high bandwidth environments. It also has the best improvement over ordinary TCP if the delay on the edge nodes (senders and receivers) is large. HS-TCP also reduces packet loss compared to ordinary TCP as shown in the figure below.



The above figure shows comparison of the rate of packet loss between HS-TCP and the ordinary TCP. HS-TCP increases the size of congestion window for faster and reliable data transfer. While MX-TCP allows the user to set a transmission rate which is used 100% hence reducing the problem of congestion and disconnection. In the event of packet loss, MX-TCP do not back off but prioritizes the lost packets to be sent again. In ad-hoc networks, MX-TCP performs better by transferring data at a constant, user specified rate. It also avoids the problem of packet loss by retransmitting the lost packets.

Hence to conclude, HS-TCP has been overall observed to work well in case of single TCP sessions with bulk data transfer over high-bandwidth links with no losses while MX-TCP is better suited over traditional TCP and HS-TCP to work in lossy, high latency environments including VSAT Links.

REFERENCES

[1] Ahuja, A., S. Agarwal, J. P. Singh, and R. Shorey. 2000. "Performance of TCP over Different Routing Protocols in Mobile Ad-Hoc Networks". IEEE VEHICULAR TECHNOLOGY CONFERENCE. 3 (Conf 51): 2315-2319.
[2] Chodorek, Agnieszka. 2008. "TCP and TCP-Friendly Protocols".
[3] Holland, G., and N. Vaidya. 2002. "Analysis of TCP Performance over Mobile Ad-hoc Networks". WIRELESS NETWORKS. 8: 275-288.
[4] Paul, http://www.stormcontrol.net/hs-tcp-vs-mx-tcp/
[5] http://www.wikinvest.com/stock/Riverbed_Technology_(RVBD)/Transport_Streamlining
[6] http://searchenterprisewan.techtarget.com/tutorial/Evaluating-Riverbed-Steelhead-WAN-optimization

BIOGRAPHIES

Aniket Deshpande is pursuing his PhD in Computer Networking at Mewar University, Chittorgarh, Rajasthan, India under the guidance of Dr. Ashok Kaushal. He received his Master of Technology in Computer Science and Engineering from Dr. MGR Educational and Research Institute, Chennai, Tamil Nadu, India in 2004. Prior to that he completed his Bachelor of Engineering in Computer Technology from Nagpur University in 2000. Network Infrastructure Optimization has been his key area of interest for last many years and is currently working with Riverbed Technology as a Solution Consultant.

Dr. Ashok Kaushal completed his Master of Technology in 1983 from Lviv Polytechnic National University, Ukraine followed by Ph.D. in 1987 from National Technical University of Ukraine (formerly Kyiv Polytechnic Institute) in Computer Technology. He has extensive Industry experience and is also presently working as a Research Guide and Visiting Faculty at Mewar University. Additionally, he is also working as a Freelance consultant with special interests in Geospatial Technologies (GIS) including recent advances in ICT, Electronics & GIS including Cloud Computing, Crowd sourcing, Big Data, Smart Phones, Digital Wallet, RFID, Mobile Computing ecosystem with Applications in e-Governance and Surveillance including Smart Cities.