ISSN (Online) 2249 - 054 X

International Journal of Computing and Corporate Research

Specialized and Refereed Journal for Research Scholars, Academicians, Engineers and Scientists

http://www.ijccr.com

International Manuscript ID : ISSN2249054X-V2I2M11-032012

VOLUME 2 ISSUE 2 March 2012

PERFORMANCE ANALYSIS OF SACK ENABLED TCP CONNECTION OVER WIRELESS LINK

Vishal Sharma [a,*], Shelly Gujral [b], Jagroop Singh [c], Jaspreet Singh[d]

[a] Shaheed Bhagat Singh College of Engineering and Technology, Ferozepur, Punjab, India
[b, c] DAVIET, Jalandhar, Punjab, India
[d] Sri Sukhmani Institute of Engineering and Technology, Dera Bassi, Punjab, India

Abstract: This work analysis the TCP connection over wireless link by evaluating selectively acknowledged data, retransmission count under different values of packet discard ratio, packet latency and max segment size supporting SACK for error control/recovery by means of OPNET simulator. Further, Further, we have also evaluated the performance of a SACK enabled TCP connection over a lossy medium for different segment sizes of ATM, FDDI, Ethernet, Frame Relay and Token Ring topologies.

***Key Words*:** TCP, Error Control mechanism, Packet Discard Ratio, Packet Latency

I. INTRODUCTION

Transmission control protocol (TCP) is a set of network standards that dominates today's communication in various networks from the wired backbone to the heterogeneous network due to its remarkable simplicity and reliability [1]. Internet protocol (IP) is a connectionless best-effort based variable length

ISSN (Online) 2249 - 054 X
International Journal of Computing and Corporate Research
Specialized and Refereed Journal for Research Scholars, Academicians, Engineers and Scientists

http://www.ijccr.com

International Manuscript ID : ISSN2249054X-V2I2M11-032012

VOLUME 2 ISSUE 2 March 2012

packet delivery network layer protocol that does not guarantee the reliable, timely, and in-order delivery of packets between end hosts but focuses on the routing mechanism based on the addressing scheme. Alternatively, it is TCP protocol that uses an end-to-end and connection-oriented packet transport mechanism to ensures the reliable and ordered delivery of data. Even then, there is probability of packet loss and out-of-ordered reception of packets that causes throughput degradation in TCP based wired/wireless networks. The dynamics of TCP can sometimes cause the network switches or routers to accumulate large queues, resulting in buffer overflows, reduced throughput and under utilization. Congestion is the main cause of packet loss in wired network with negligible impact of random bit error rate (BER). To eradicate such failures, TCP implements flow control and congestion control algorithms based on the sliding window and additive increase multiplicative decrease (AIMD) algorithms [2]. TCP Reno is one of the most widely adopted TCP reactive congestion control schemes having four transmission phases as slow start, congestion avoidance, fast recovery, and fast retransmit [3]. TCP maintains two variables such as the congestion window size (cwnd), which is initially set to be 1 maximum segment size (MSS), and the slow start threshold (ssthresh). At the beginning of a TCP connection, the sender enters the slow start phase, in which cwnd is increased by 1 MSS for every ACK received; thus, the TCP sender's cwnd grows exponentially in round-trip times (RTTs). When cwnd reaches ssthresh, the TCP sender enters the congestion avoidance phase. Reno employs a sliding-window-based flow control mechanism allowing the sender to advance the transmission window linearly by one segment upon reception of an ACK, which indicates the last in-order packet 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 (DUPACKs), or the sender's retransmission timeout (RTO) timer expires. These events activate TCP's fast retransmit and recovery, by which the sender reduces the size of its congestion window (cwnd) to half and linearly increases cwnd as in congestion avoidance, resulting in a lower transmission rate to relieve the link congestion. After one lost packet is recovered, Reno terminates the fast recovery mechanism. Due to the nature of wireless networks, correlated errors may induce multiple

ISSN (Online) 2249 - 054 X
International Journal of Computing and Corporate Research
Specialized and Refereed Journal for Research Scholars, Academicians, Engineers and Scientists
http://www.ijccr.com
International Manuscript ID : ISSN2249054X-V2I2M11-032012
VOLUME 2 ISSUE 2 March 2012

packet drops. Therefore, the Reno scheme would be forced to invoke multiple fast recovery procedures back and forth, slowing down recovery of the lost packet. New Reno [4] modifies the fast recovery mechanism of Reno to cope with multiple losses from a single window; this is one of the characteristics of wireless networks, where a fading channel may cause contiguous packet loss. In New Reno the fast recovery mechanism does not terminate until multiple losses, indicated by the reception of partial ACKs, from one window are all recovered. The limitation of New Reno is, however, that it cannot distinguish the cause of the packet loss; thus, a more effective fast recovery algorithm cannot be implemented. TCP SACK [5] is a selective ACK (SACK) option for TCP, targeting the same problem Reno tries to tackle. While the feedback of Reno and New Reno is based on cumulative ACKs, SACK employs a selective repeat retransmission policy.

An analytical model for the TCP steady state throughput as a function of the network utilization factor, round trip time and packet drop rate for unlimited data transfer to capture the fast retransmit and recovery mechanisms has proposed over wireless links with a modified fast retransmit algorithm that allows packet recovery with smaller congestion window sizes than possible with TCP Reno or NewReno, thereby reducing the likelihood of timeouts [6]. In wireless networks, a retransmission timeout is inevitable, when the retransmission of a lost packet fails to reach the destination particularly due to changing level of congestion and bit error rate in wireless channel [7]. To prevent these unwanted retransmissions, congestion control as well as error recovery are implemented by a sliding window to modify the present SACK implementation [8]. The proposed implementation of SACK mechanism increases the throughput of SACK enabled TCP connections [9, 10].

II. SIMULATION SETUP

Using OPNET simulator, we have designed an IEEE 802.11 WLAN based simulation scenario to configure a TCP connection based application. The Hyper Text Transfer Protocol (HTTP) is built in the workspace by means of HTTP Application. An IP-capable 32 serial links WAN is used to connect TCP-client and TCP server by means of point-to-point(PPP) enabled DS1 link (1.5Mbps) with different packet discard ratio and

ISSN (Online) 2249 - 054 X
International Journal of Computing and Corporate Research
Specialized and Refereed Journal for Research Scholars, Academicians, Engineers and Scientists
http://www.ijccr.com
International Manuscript ID : ISSN2249054X-V2I2M11-032012
VOLUME 2 ISSUE 2 March 2012

packet latency (round-trip delay). The TCP parameters of HTTP Client attribute and set the Selective ACK (SACK) attribute to enable. Set the Fast Retransmit and Fast Recovery attributes to disabled. Fast Retransmit and Fast Recovery are alternate TCP error control mechanisms. By disabling them, we can concentrate on the effects of the SACK mechanism. The TCP parameters of HTTP server attribute and set the Maximum Segment Size (bytes) attribute to 512. This will ensure that each TCP packet sent 512 bytes long data. In order to correct for segments that have been lost or corrupted, TCP buffers data at the source does retransmissions when necessary. Typically, losses are detected at the source via a timeout mechanism. If a segment is sent and an acknowledgement is not received before the timer expires, the segment is retransmitted. By default, TCP relies on cumulative acknowledgements. A cumulative acknowledgement for byte **x** acknowledges all bytes less than **x**. Since many segments are sent at once, it is often not clear to the TCP source which segments have been lost. As a result, the TCP source retransmits all unacknowledged segments when a loss is detected (a go-back-n strategy). This strategy may be inefficient if most of the segments actually did arrive at the destination. An enhancement to TCP to deal with this inefficiency is the Selective Acknowledgment (SACK) option. If both the TCP source and destination agree to use this option, selective acknowledgements are used rather than cumulative. With this specific information regarding loss, the TCP source can do selective repeat retransmissions, i.e., retransmit only the lost segments, rather than all unacknowledged segments.

III. RESULT & DISCUSSION

To evaluate the overall performance of TCP connection, we have determined the various parameters of IEEE 802.11 WLAN such as selectively acknowledged data, retransmission count under different values of packet discard ratio, packet latency and max segment size supporting SACK for error control/recovery. Figure 1 show that selectively acknowledged data of a TCP connection of packet latency of 0.25 decreases with increasing packet discard ratio and it is concluded that with less packet latency and packet discard ratio, the performance of a SACK enabled TCP connection over a lossy medium is better as shown in Figure 1 and 2 in terms of selectively acknowledged data. Further, we have also evaluated

ISSN (Online) 2249 - 054 X
International Journal of Computing and Corporate Research
Specialized and Refereed Journal for Research Scholars, Academicians, Engineers and Scientists

http://www.ijccr.com

International Manuscript ID : ISSN2249054X-V2I2M11-032012

VOLUME 2 ISSUE 2 March 2012

the performance of a SACK enabled TCP connection over a lossy medium for different segment sizes of ATM, FDDI, Ethernet, Frame Relay and Token Ring technologies as shown in Figure 3 with packet discard ratio of 1%, packet latency of 0.25%.

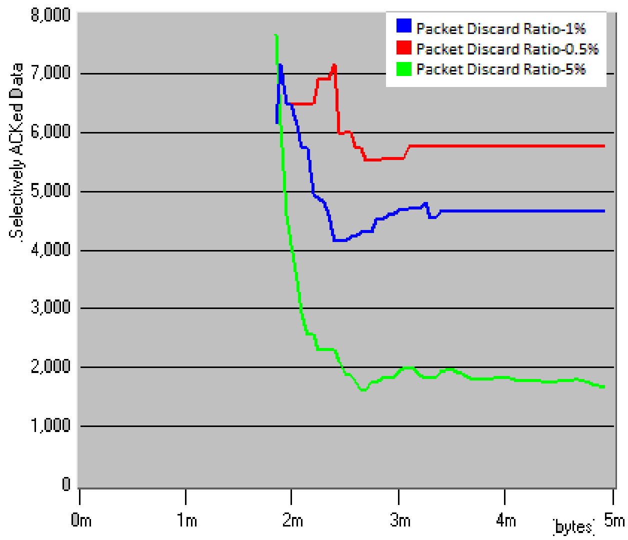


Figure 1 Selectively acknowledged data for different values of Packet Discard Ratio with packet Latency of 0.25%, max segment size of 512 and sack is enabled only.

It is shown that when sack is enabled and fast retransmit, fast recovery disabled SACK enabled TCP connection over a lossy medium performs better with Ethernet and FDDI than that of other technologies in terms of selectively acknowledged data. Figure 4 and 5 shows that retransmission count of a SACK enabled TCP connection over a lossy medium with different packet latency and packet discard ratio and shows that retransmission counts increases with increases packet discard ratio and with less value of packet latency with maximum segment size of 512 Mbps.

ISSN (Online) 2249 - 054 X

International Journal of Computing and Corporate Research

Specialized and Refereed Journal for
Research Scholars, Academicians, Engineers and Scientists

http://www.ijccr.com

International Manuscript ID : ISSN2249054X-V2I2M11-032012

VOLUME 2 ISSUE 2 March 2012

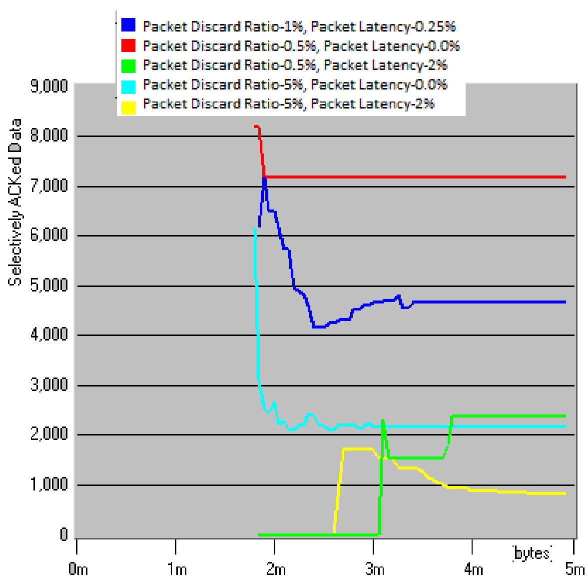


Figure 2 Selectively acknowledged data for different values of Packet Discard Ratio and Packet Latency with max segment size-512 and sack is enabled only.

ISSN (Online) 2249 - 054 X

International Journal of Computing and Corporate Research

Specialized and Refereed Journal for Research Scholars, Academicians, Engineers and Scientists

http://www.ijccr.com

International Manuscript ID : ISSN2249054X-V2I2M11-032012

VOLUME 2 ISSUE 2 March 2012

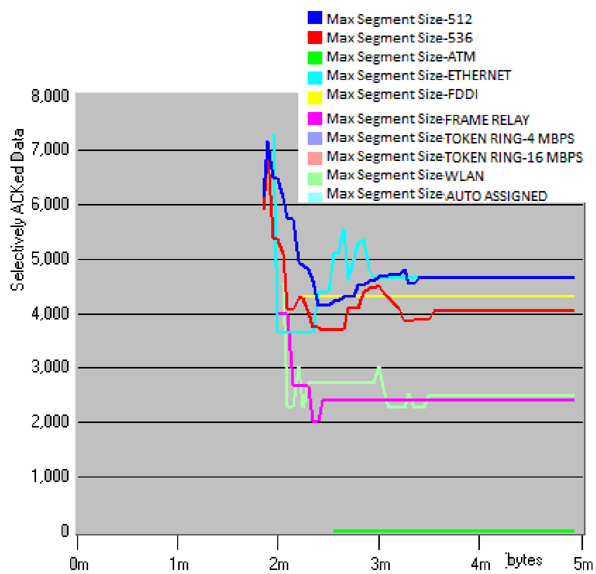


Figure 3 Selectively Acknowledged data for different values of Max Segment Size, when packet discard ratio of 1%, packet latency of 0.25%, and sack is enabled.

ISSN (Online) 2249 - 054 X

International Journal of Computing and Corporate Research

Specialized and Refereed Journal for Research Scholars, Academicians, Engineers and Scientists

http://www.ijccr.com

International Manuscript ID : ISSN2249054X-V2I2M11-032012

VOLUME 2 ISSUE 2 March 2012

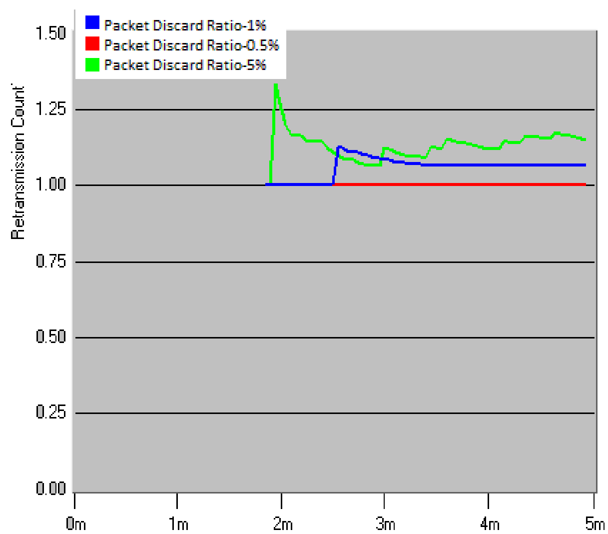
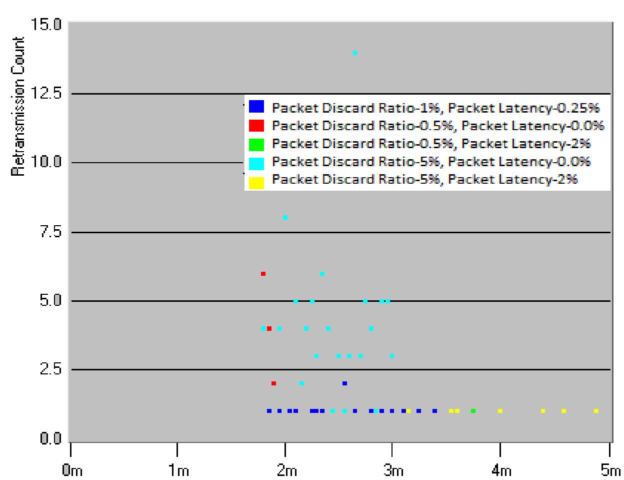


Figure 4 Retransmission Count for different values of Packet Discard Ratio with packet Latency of 0.25%, max segment size of 512 and sack is enabled only



ISSN (Online) 2249 - 054 X

International Journal of Computing and Corporate Research

Specialized and Refereed Journal for Research Scholars, Academicians, Engineers and Scientists

http://www.ijccr.com

International Manuscript ID : ISSN2249054X-V2I2M11-032012

VOLUME 2 ISSUE 2 March 2012

Figure 5 Retransmission Count for different values of Packet Discard Ratio and Packet Latency, max segment size-512, and sack is enabled only

CONCLUSION

In this paper, the performance analysis of a Sack enabled TCP connection over wireless link under different values of packet discard ratio, packet latency and max segment size is reported by evaluating selectively acknowledged data and retransmission count by means of OPNET simulator. From our simulative results, it is concluded that by keeping packet latency and packet discard ratio low, the Sack enabled TCP connection over wireless link performs better along with Ethernet and FDDI topologies.

REFERENCES

[1] Ye Tian Kai Xu Ansari, "TCP in wireless environments: problems and solutions", IEEE Communications Magazine, vol. 43, March 2005

[2] D. Chiu and R. Jain, "Analysis of the Increase/Decrease Algorithms for Congestion Avoidance In Computer Networks," J. Comp. Net., vol. 17, no. 1, June 1989, pp. 1–14.

[3] K. Fall and S. Floyd, "Simulation-Based Comparisons of Tahoe, Reno and SACK TCP," Comp. Commun. Rev., vol. 26, no. 3, July 1996, pp. 5–21

[4] S. Floyd and T. Henderson, "The New Reno Modification to TCP's Fast Recovery Algorithm," RFC 2582, Apr. 1999.

[5] M. Mathis et al., "TCP Selective Acknowledgment Options," RFC 2018, Oct. 1996.

ISSN (Online) 2249 - 054 X
International Journal of Computing and Corporate Research
Specialized and Refereed Journal for Research Scholars, Academicians, Engineers and Scientists

http://www.ijccr.com

International Manuscript ID : ISSN2249054X-V2I2M11-032012

VOLUME 2 ISSUE 2 March 2012

[6] Jayananthan, A. Sirisena, H. Garg, “Analytical Model of TCP with Enhanced Recovery Mechanism for Wireless Environments”, IEEE International Conference, 24-28 June 2007.

[7] Prasanthi, S. Sang-Hwa Chung Changwoo Ahn, “An Enhanced TCP Mechanism for Detecting and Differentiating the Loss of Retransmisssions over Wireless Networks”, Advanced Information Networking and Applications (AINA), IEEE International Conference, 22-25 March 2011.

[8] Aweya, J. Ouellette, M. Montuno, D.Y. Yao, “Enhancing network performance with TCP rate control” Global Telecommunications Conference, 2000” Vol. 3, 2000

[9] Srijith, K.N. Jacob, L. Ananda, A.L., “Worst-case performance limitation of TCP SACK and a feasible solution”, Communication Systems, 2002. ICCS 2002. The 8th International Conference, vol. 2, 25-28 Nov. 2002

[10] Bournas, R. M. , “Optimization of TCP segment size for file transfer” IBM Journal of Research and Development , vol. 41, May 1997