Webpage: www.ijaret.org
Volume 2, Issue VIII Aug 2014
ISSN 2320-6802

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY

WINGS TO YOUR THOUGHTS.....

An Enhanced TCP Congestion Control Approach to Evaluate the Performance over WiMAX Network

Manju[1], Lovnish Bansal[2]

[1]Bhai Maha Singh College of Engineering & Tehnology, Punjab Technical University,
Kotkapura Road, Sri Muktsar Sahib (Punjab)
rathi.ancester@gmail.com

[2] Bhai Maha Singh College of Engineering & Tehnology, Punjab Technical University,
Kotkapura Road, Sri Muktsar Sahib (Punjab)
lovnishbansal@gmail.com

Abstract*: WiMax (World Wide Interoperability for Microwave access) technology is presently one of the most promising global telecommunication systems. Great hopes and important investments have been made for WiMax, which is a Broadband Wireless Access System having many applications: fixed or last-mile wireless access, backhauling, mobile cellular network, telemetering, etc. Wimax is based on the IEEE 802.16 standard, having a rich set of features. It provides high-speed access to the Internet where the Transmission Control Protocol (TCP) is the core transport protocol. Congestion control is widely considered to be a key problem for WiMax, which can lead to degradation in the quality of service (QoS) of the WiMax network. Over the years, numerous Transmission Control Protocol (TCP) congestion control algorithms have been proposed for deployed wireless and wired networks. The main aim of these algorithms was to successfully handle the congestion and minimizing packet loss. We concentrate on three congestion control algorithms (TCP Variants). These TCP Variants include TCP-Tahoe, TCP-Sack and TCP-Fack. So in this thesis work an extensive experimental analysis using Network Simulator (NS-2) has been carried out to study the effect of higher offered load on the base station with different TCP Variants in relation with RED (Random Early Detection) an active queue management technique, which ensures the the provision of Quality of Service (QoS) in terms of Throughput, Packet Delivery Ratio, Average Delay and Routing Overhead.*

Keywords: *WiMax, TCP, TCP-Sack, TCP-Fack, QoS, NS-2.*

1. INTRODUCTION

Recently, WiMAX, a standards-based technology, has emerged as an alternative to cable and DSL in delivering last-mile wireless broadband access [3]. The mobile WiMax physical layer (PHY) is based on Orthogonal Frequency Division Multiplexing (OFDM) [5].Wimax is based on the IEEE 802.16 standard, having a rich set of features. It provides high-speed access to the Internet where the Transmission Control Protocol (TCP) is the core transport protocol. Transmission control protocol (TCP) [1], [2] is the predominant Internet protocol and carries approximately 90% of Internet traffic in today's heterogeneous wireless and wired networks. Congestion control is widely considered to be a key problem for WiMax. The congestion occurs when there is a lot of traffic in the networks [7]. The Internet is no exception. Congestion can be dealt with by employing a principle borrowed from physics - the law of conservation of packets. The idea is to refrain from injecting a new packet into the network until an old one leaves. TCP attempts to achieve this goal by dynamically manipulating the window size. This paper is organised as follows: Section II reviews the following TCP variations: Tahoe, Reno, New Reno, Vegas, Sack and Fack; Section III describes the test environment for simulating the WiMAX networks; Section IV presents the simulation results and the last section draws our conclusions.

2. LITERATURE REVIEW

Mechanisms devised to control congestion window are slow start, Additive Increase and Multiplicative Decrease (AIMD), congestion avoidance, fast retransmit, and fast recovery. Various TCP flavors employ these congestion control algorithms to regulate the sending rate as a function of perceived congestion [9].

TCP Tahoe maintains two variables that is a congestion window and a slow-start threshold as shown in figure 1. The congestion window determines the number of segments that is transmitted within an RTT. At the start of a TCP session, the congestion window is set to 1, and the transmitter sends only one segment and waits for an acknowledgment. When an ACK is received, the congestion window is doubled, and two segments are transmitted at a time. This process of doubling the congestion window continues until it reaches the maximum indicated by the advertised window size. At this point, TCP infers that the network is congested and begins the recovery process by dropping the congestion window back to one segment. Resetting the congestion window to one

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS.....

segment allows the system to clear all packets in pipeline. Thus every time a packet is lost it waits for a timeout and the pipeline is emptied.

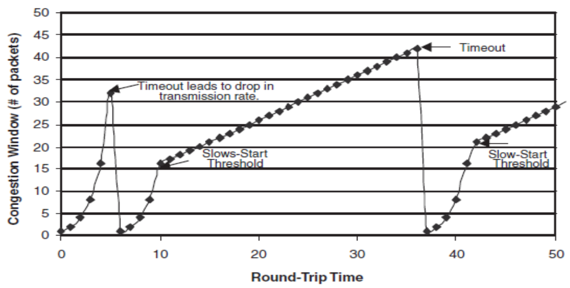

Figure 1 TCP congestion control

TCP Reno retains the basic principle of Tahoe, but uses the logic of duplicate acknowledgements (dupacks) to trigger Fast Retransmit. After 3 dupacks, TCP Reno takes it as a sign of segment lost and retransmit the packet immediately and enter Fast Recovery. In Fast Recovery, ssthresh and cwnd is set to half the value of current cwnd. For each subsequent dupack, increase cwnd by one and transmit a new segment if the new value permits it. TCP Reno cannot detect multiple packet loss within the same window [4].

TCP New Reno [4] is a slight modification over TCP-RENO. It is able to detect multiple packet losses and thus is much more efficient that RENO in the event of multiple packet losses. Like RENO, New-RENO also enters into fast-retransmit when it receives multiple duplicate packets, however it differs from RENO in that it does not exit fast-recovery until all the data which was out standing at the time it entered fast recovery is acknowledged.

TCP Vegas is a TCP implementation which is a modification of RENO. It builds on the fact that proactive measure to encounter congestion is much more efficient than reactive ones. It tried to get around the problem of coarse grain timeouts by suggesting an algorithm which checks for timeouts at a very efficient schedule. Also it overcomes the problem of requiring enough duplicate acknowledgements to detect a packet loss, and it also suggests a modified slow start algorithm which prevents it from congesting the network.

TCP Sack is an extension of TCP RENO and it works around the problems face by TCP RENO and TCP New-RENO, namely detection of multiple lost packets, and re-transmission of more than one lost packet per RTT. SACK retains the slow-start and fast retransmits parts of RENO. It also has the coarse grained timeout of Tahoe to fall back on, in case a packet loss is not detected by the modified algorithm. SACK algorithm allows a TCP receiver to acknowledge out-of-order segments selectively rather than cumulatively by acknowledging the last correctly in order received segment [9]. If there are no such segments outstanding then it sends a new packet**.** Thus more than one lost segment can be sent in one RTT.

Finally, **TCP Fack** is a special algorithm that works on top of the SACK options, and is geared at congestion controlling. FACK algorithm uses information provided by SACK to add more precise control to the injection of data into the network during recovery – this is achieved by explicitly measuring the total number of bytes of data outstanding in the network [8]. FACK decouples congestion control from data recovery thereby attaining more precise control over the data flow in the network. The main idea of FACK algorithm is to consider the most forward selective acknowledgement sequence number as a sign that all the previous acknowledged segments were lost. This observation allows improving recovery of losses significantly.

3. SIMULATION ENVIRONMENT

In this section we will present the test setup used for comparing the above TCP schemes. The traffic scenarios were implemented in network simulator-2 as shown in figure 2. Network Simulator-2 (NS-2) is an open source, discrete event network simulator [6]. Table 1 depicts the most important WiMAX and traffic parameters used in our simulations.

Table 1 Simulation Parameters

Parameter	**Value**
Simulation time	200 sec.
Simulation area	1000m x 1000m
Antenna	Omni antenna
Number of nodes	30
TCP-variants	TCP-Tahoe, TCP-Sack, TCP-Fack
Packet size	512 bytes
Traffic	FTP
Routing protocol	OLSR
Mobility model	Random waypoint model
Buffer management technique	Random early detection(RED)

A detail simulation model based on NS-2 has been used in the evaluation, and in order to perfectly evaluate the effect of out-of-order packet while some of the TCP variants. The source-destination pairs are spread instant over the network. The data generator is FTP. Mobility models were created for the simulations using 30 nodes as shown in figure, and this model was set in such a way that first all the 30 nodes were provided with initial location in the given rectangular topography field. The field configuration used is: 1000 m x 1000 m field. Then all the nodes move within their boundary by setting their final destination and the speed that each node move with. All the simulations are run for 200 simulated seconds. Different mobility and identical traffic scenarios are used across the protocol to collect fair results.

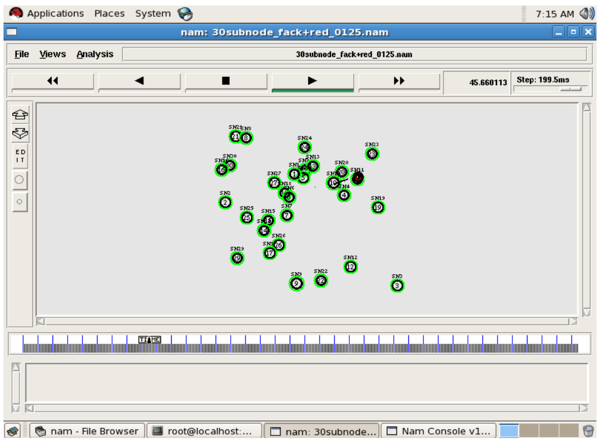

Figure 2 Nam animation trace with node deployment

4. SIMULATION RESULTS

In this section we present our simulation scenarios in WiMAX and discuss the results obtained. With the help of 2D graphs, the simulation has been analyzed for various TCP variants based on traffic generators like FTP under higher offered load network scenario based on cyclic prefixes using NS-2.

4.1 Packet Delivery Ratio (PDR)

Figure 3 shows the packet delivery ratio for TCP-Tahoe, TCP-Sack, TCP-Fack and TCP-Fack+RED when the cyclic prefix is varied. Simulation results shows that TCP-Fack gives higher performace when the value of cyclic prefix decreases. It is observed that the packet delivery ratio of TCP FACK with buffer management technique namely RED (TCP-FACK+RED) over OLSR under higher offered load is better than both TCP-Tahoe and TCP-Sack.

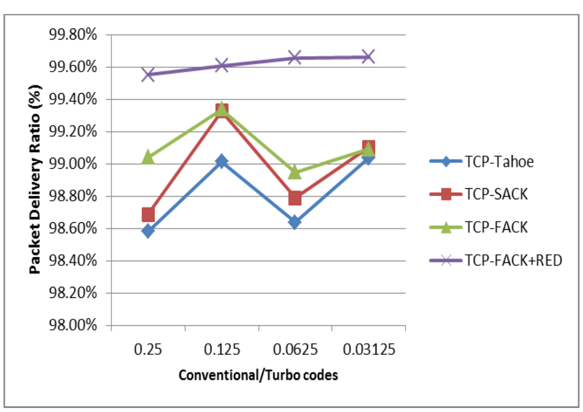

Figure 3 Impact of Coventional/Turbo Codes on the Packet Delivery Ratio under Different TCP-Variants with RED

4.2 Throughput

The figure 4 shows the impact of mean speed on the throughput. It is observed that the throughput of TCP-FACK+ RED is better than other two TCP-variants i.e TCP-Tahoe and TCP-Sack. Throughput is maximum for cyclic prefix 0.25. The throughput is representative of number of bits received per second.

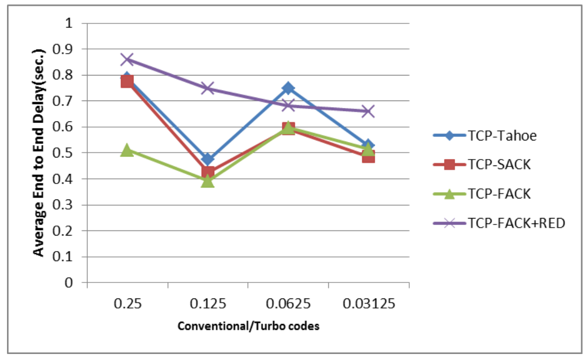

Figure 4 Impact of coventional/Turbo codes on Throughput under different TCP-Variants with RED

4.3 Average Delay

The figure 5 shows the end-to-end delay when the number of source destination pairs are maximum. OLSR protocol uses the route cache which many a times contains stale routes, as a result delay is comparatively higher. As the load on base satation increases the delay increases. The end to end delay of TCP-FACK+RED is higher than other TCP-variants i.e TCP-Tahoe and TCP-Sack.

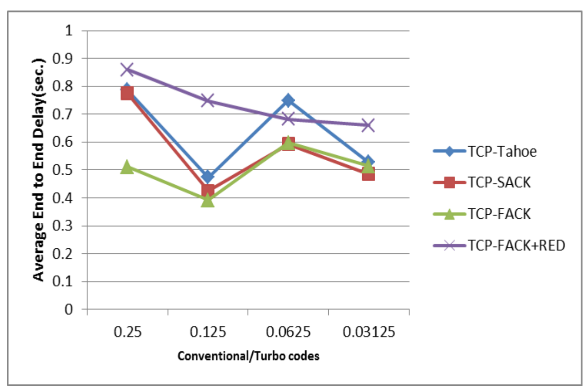

Figure 5 Impact of coventional/Turbo codes on the Average Delay under different TCP-Variants with RED

4.4 Routing Load

The figure 6 shows the impact of cyclic prefix on the routing load. It is observed that the routing load of TCP-FACK+RED is better than both TCP-Tahoe and TCP-Sack variants. When cyclic prefix is lower the value of routing load is lower as cyclic prefix increases routing load increases.

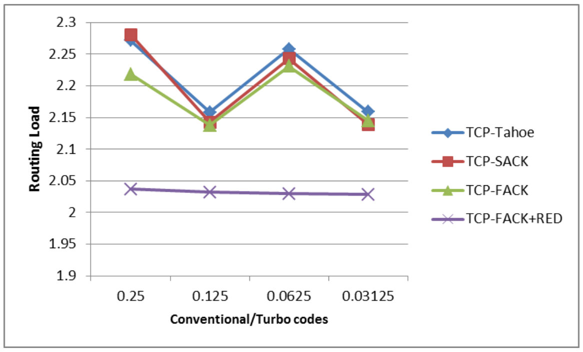

Figure 6 Impact of coventional/Turbo codes on the Routing Load under different TCP-Variants with RED

5. CONCLUSIONS AND FUTURE WORK

It is a well known fact that TCP can experience significant performance degradation during hand-off, if multiple packet droppings, packet re-ordering or exorbitant hand-off delays occur. We have shown that the reaction on packet droppings and re-ordering is very much related to the implemented TCP version. Different TCP versions react with different types of behavior. In addition, from the perspective of transport layer, we believe that TCP will be on top of the routing protocols for reliable data transmission. We simulated and compared the performance of various TCP algorithms viz. TCP-Tahoe, TCP-Sack, TCP-Fack in WiMax network. Simulation results indicated that in WiMax network with higher congestion at the node, TCP-Fack+RED outperforms other two congestion control algorithms i.e TCP-Tahoe and TCP-Sack. The Throughput, Packet Delivery Ratio and Routing Load in case of TCP-Fack+RED is higher than the remaining two TCP variants i.e TCP-Tahoe and TCP-Sack. But TCP-Fack+RED shows significant performance degradation (i.e increase in delay parameter) in case of Average Delay. Through the extensive simulations, we found that which to select among TCP variants along with queue management schemes in order to control congestion at the node.

On the basis of the results obtained from simulation graphs and some trials in the literature, we can develop a bandwidth estimation technique to improve TCP-FACK performance over WiMAX environment, which is our interesting future work.

References

[1] Adhicandra, I., Garroppo, R. G. and Giordano, S. (2009), "Optimizing System Capacity and Application Delays in WiMAX Networks", Wireless Communication Systems IEEE, Print ISBN: 978-1-4244-3584-5, pp.540-544.

[2] Alam, M. S. and Ahmmed, K. T. (2012), "Performance Analysis of Fixed WiMAX in Metropolitan Area", International Journal of Electronics and Communication Engineering, Volume 5, Number 3 , pp. 331-341.

[3] Behmann, F. (2005), "Impact of Wireless (Wi-Fi, WiMAX) on 3G and Next Generation — An Initial Assessment", Electro Information Technology, 2005 IEEE International Conference on, pp. 1 – 6, ISBN:0-7803-9232-9.

[4] Francis, B., Narasimhan, V., Nayak, A. and Stojmenovic, I. (2012), "Techniques for Enhancing TCP Performance in Wireless Networks", International Conference on Distributed Computing Systems Workshops IEEE, DOI 10.1109/ICDCSW, pp. 222-230.

[5] Garroppo, R. G., Giordano, S., Iacono, D. and Tavanti, L. (2011), "Game Theory and Time Utility Functions for a Radio Aware Scheduling Algorithm for WiMAX Networks", Journal Wireless Networks, Vol. 17, Issue 6, pp. 1441-1459.

[6] Khan, A. U. R., Bilal, S. M. and Othman, M., "A Performance Comparison of Network Simulators for Wireless Networks", pp. 1-6.

[7] Kirov, G. (2005), "A Simulation Analysis of the TCP Control Algorithms", International Conference on Computer Systems and Technologies – CompSysTech, pp. - IIIB.14-1 -IIIB.14-6.

[8] Mathis, M. and Mahdavi, J. (1996), "Forward Acknowledgment: Refining TCP Congestion Control", Copyright 1996 by Association for Computing Machinery Inc ACM To appear in Computer Communication Review a publication of ACM SIGCOMM volume 2. number 4.

[9] Subedi, L., Najiminaini, M. and Trajković, L. (2009), "Performance E Evaluation of TCP Tahoe, Reno, Reno with SACK, and NewReno Using OPNET Modeler", Proceeding of International Conference on Methods and models in Computer Science IEEE.