Volume 4, Issue 5, May 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Performance Evaluation of Different TCP Variants Over Wimax Environment

Ashish Chhabra
Dept. of Computer Science and Engg.
MVVEC,Jagadhri, India

Mauli Joshi
Dept. of Computer Science and Engg.,
MVVEC,Jagadhri, India

Shanu Malhotra
Dept. of Computer Science and Engg.,
ISTK,Kalwad, India

Abstract— ***WiMAX 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. This standard defines the Medium Access Layer and the Physical Layer of a fixed and mobile Broadband Wireless Access System. WiMAX to provide high-speed access to the Internet where the transmission control protocol (TCP) is the core transport protocol. Unlike routing, where packets are relayed hop-by-hop toward their destination, TCP actually provides reliable end to-end transmission of transport-level segments from source to receiver. As TCP was designed for wired networks it considers that all packet loss in the network is due to congestion. Wireless medium is more exposed to transmission errors and sudden topological changes. So in this paper, we investigate the effects of subscriber's mean speed on the performance characteristics of three representative TCP schemes, namely TCP-New Reno, Westwood and Cubic, in WiMAX networks, under the conditions of correlated lossy links, route failures and network congestion using ns2.***

Keywords— ***WiMAX, IEEE 802.16, TCP***

I INTRODUCTION

Since the final decades of the twentieth century, data networks have known steadily growing success. After the installation of fixed Internet networks in many places all over the planet and their now large expansion, the need is now becoming more important for wireless access. There is no doubt that by the end of the first decade of the twentieth century, high-speed wireless data access will be largely deployed worldwide. A large number of wireless transmission technologies exist, other systems still being under design. These technologies can be distributed over different network families, based on a network scale. In fig. 1, a now-classical representation is shown of wireless network categories, with the most famous technologies for each type of network.

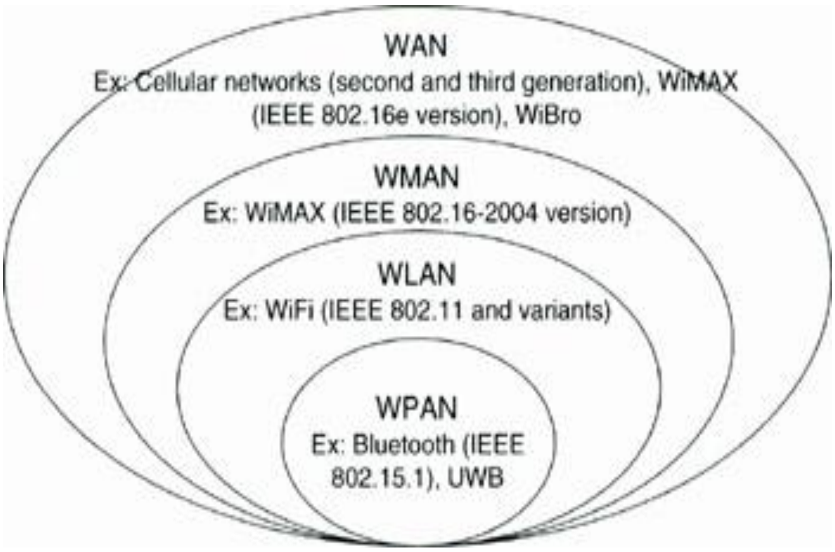

Fig. 1 : Illustration of network types

Obviously, the traditional mechanisms such as wifi, Bluetooth etc., to access network are no longer suitable and can not meet the newly upcoming requirement. Fortunately, the Broadband Wireless Access (BWA) technology (such as WiMax), which can meet people's need to access the Internet conveniently, becomes more popular in recent years.

A. WiMAX

WiMAX stands for Worldwide Interoperability for Microwave Access. It is a broadband wireless point-to-multipoint specification from the IEEE 802.16 working group. It is a telecommunications protocol that provides fixed and mobile

Internet access. Although initial WiMAX deployments are likely to be for fixed applications, the full potential of WiMAX will be realized only when used for innovative nomadic and mobile broadband applications.

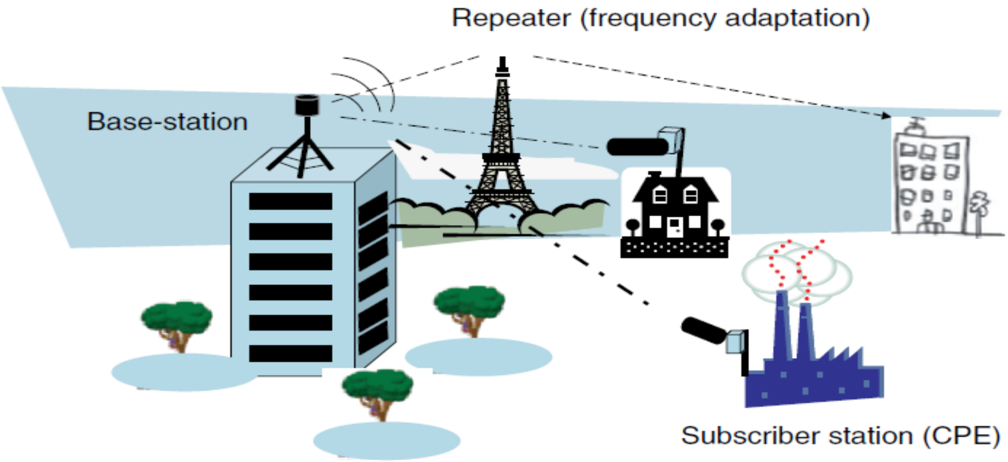

Fig. 2: IEEE 802.16 fixed deployment scenario.

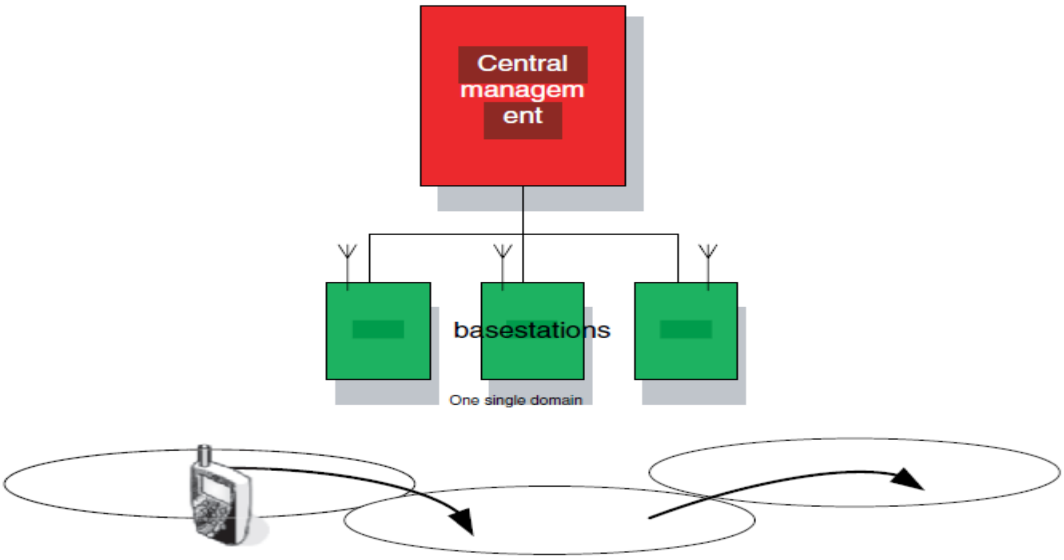

Fig. 3: IEEE 802.16 mobile deployment scenario.

WiMAX technology in its IEEE 802.16e-2005 incarnation will likely be deployed by fixed operators to capture part of the wireless mobility value chain in addition to plain broadband access. As endusers get accustomed to high-speed broadband at home and work, they will demand similar services in a nomadic or mobile context, and many service providers could use WiMAX to meet this demand. For WiMAX, to be able to realize the objective of high speed Internet access, it must effectively support the core transport protocol of the Internet that is Transmission Control Protocol (TCP). Standard TCP congestion control is based on the reduction of its congestion window after a packet loss. Although such behavior works fairly well in the wired networks, where packets losses are almost always caused by link congestion, it becomes rather inefficient when used for data transport in WiMAX networks. In the wireless environment the possible reasons of packet loss include fading, temporary disconnections, and handovers. Even when some losses are compensated in Data Link Layer, a part of them still appears in Transport Layer for high Bit Error Rates (BER).

B. **TCP-Congestion Control**

When the load offered to any network is more than it can handle, congestion builds up. 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 (i.e., is delivered). TCP attempts to achieve this goal by dynamically manipulating the window size. The first step in managing congestion is detecting it. In the old days, detection of congestion was difficult. A timeout caused by a lost packet could have been caused by either noise on a transmission line or packet discard at a congested router. Fig. 4 illustrate how TCP manage network congestion. TCP maintains two variables that is a congestion window and a slow-start threshold.

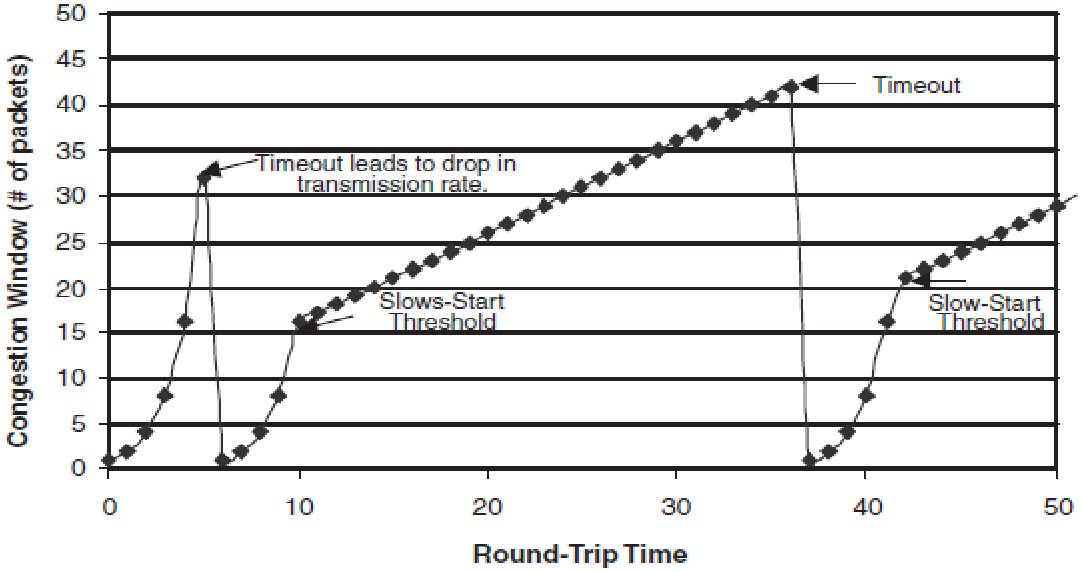

Figure 4: TCP congestion control

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 or until the sender fails to get an acknowledgment before the timer expires. 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 segment allows the system to clear all packets in transit. Now, if a retransmission also fails, the TCP sender will also exponentially back off its retransmission time, providing more time for the system to clear the congestion. If transmission is successful after restart, the process of doubling the congestion window size after every transmission continues until the contention window size reaches half the size at which it detected the previous congestion. This is called the *slow-start threshold*. Network congestion may also be detected by receiving more than two or three duplicate ACK packets, which are sent when packets are received out of order. When that happens, TCP performs a fast retransmit the missing packet without waiting for the timeout to expire and fast recovery that is follow the congestion-avoidance mechanism without resetting the congestion window back to 1.

Clearly, TCP provides a mechanism for reliable end-to-end transmission without requiring any support from intermediate nodes. This is done by making certain assumptions about the network. Specifically, TCP assumes that all packet losses, or unacknowledged packets and delays are caused by congestion and that the loss rate is small. This assumption is not valid in a *wireless network*, where packet errors are very frequent and caused mostly by poor channel conditions. Responding to packet errors by slowing down does not solve the problem if the errors are not caused by congestion. Instead, it serves only to unnecessarily reduce the throughput. Frequent errors will lead to frequent initiation of slow-start mechanisms, keeping TCP away from achieving steady state throughput.

C. **TCP-Variants**

The TCP protocol has been extensively tuned to give good performance at the transport layer in the traditional wired network environment. However, TCP in its present form is not well suited for ad hoc networks where packet loss due to broken routes can result in the counterproductive invocation of TCP‘s congestion control mechanisms.

1. **TCP-Newreno**

TCP New Reno maintains two variables, the congestion window size (*cwnd*), which initially set to 1 segment, and SS Threshold (*ssthresh*). At the beginning of the TCP connection, the sender enters the *Slow Start (SS)* phase, in which it increases the *cwnd* by 1 segment for every ACK it receives. When *cwnd* reaches the ssthresh, the TCP sender enters the *Congestion Avoidance (CA)* phase, in which it increases the *cwnd* by 1/*cwnd* for every ACK it receives, in order to slowly probe the available network bandwidth. This linear growth ends when *cwnd* reaches the receiver's advertized window, or by the reception of 3 DUPACKs. In the latter case, TCP infers that packets were lost due to link congestion, and it reduces the *cwnd* by ½ of its current value, in an attempt to prevent network collapse *(Fast Recovery).* This *Additive Increase Multiplicative Decrease (AIM.).* TCP's reactive congestion control and avoidance mechanism was proved incapable of handling efficiently mixed-type packet losses happening in wired/wireless heterogeneous networks [16].

2. **TCP-Westwood**

TCP Westwood makes no attempt to correct the problem of non-congestion packet loss in wireless networks solely like Veno, but rather to improve the efficiency of TCP in all heterogeneous networks. It estimates the network's bandwidth by properly low-pass filtering and averaging the rate of returning acknowledgment packets per RTT. It then uses this bandwidth estimate to adjust the *ssthresh* and the *cwnd* to a value close to it when a packet loss is experienced *(adaptive decrease).* In particular, when three DUPACKs are received, both the *cwnd* and *ssthresh* are set equal to the *Estimated*

Bandwidth (BWE) times the minimum measured RTT *(RTTmin)*; when a coarse timeout expires, the *ssthresh* is set as before, while the *cwnd* is set equal to one. The improvement of Westwood is a more realistic bandwidth estimation in comparison to TCP Vegas, which significantly increases TCP throughput over wireless links. TCP Westwood has also been tested in against handovers in simulated [16].

3. **TCP-Cubic**

CUBIC is an enhanced version of BIC: it simplifies the BIC window control and improves its TCP-friendliness and RTT-fairness. The window growth function of CUBIC is governed by a cubic function in terms of the elapsed time since the last loss event. TCP-cubic function provides a good stability and scalability. Furthermore, the real-time nature of this transport protocol keeps the window growth rate independent of RTT, which keeps the protocol TCP friendly under both short and long RTT paths..[12]

II. RELATED WORK

Georgi Kirov [1], focuses on the different congestion control mechanisms implemented by the Transmission Control Protocol (TCP). The author presents an experimental estimation of the TCP control algorithms: Slow-Start and Congestion Avoidance without Fast Retransmit, Tahoe that includes Fast Retransmit and Fast Recovery, and Reno using a modified version of the Fast Recovery. The TCP performance analysis is based on different scenarios of the network simulation with low percentages of the packet loss. The results for Reno are slightly better than Tahoe. The advantage of the Reno algorithms in comparison with Tahoe one is when packet loss is detected, the window size is reduced to one half of the current window size and the congestion avoidance, but not slow start is performed.

K. Tsiknas et al. [2], evaluate through simulations the performance characteristics of various TCP schemes namely - TCP New Reno, Vegas, Veno, Westwood and BIC, in WiMAX networks, by taking into account the effects of wireless channel errors, link congestion in both forward and They also suggest Binary Increase Adaptive Decrease (BIAD) paradigm will be benefited by both the quick window expansions of BIC and by the appropriate window adaptations of Westwood, thus offering in overall a better performance in WiMAX networks.

Gerla et al. [3], investigated the impact of the MAC protocol on performance of TCP on multi-hop networks. Chandran et al. [4] proposed the TCP-Feedback (TCP-F) protocol, which uses explicit feedback in the form of route failure and reestablishment control packets. Performance measurements were based on a simple one-hop network, in which the link between the sender and receiver failed/recovered according to an exponential model. Also, the routing protocol was not simulated.

Md. Shohidul Islam et al. [5], focuses on analysis of eleven variants-Tahoe, Full-Tcp, TCP-Asym, Reno, Reno-Asym, Newreno, Newreno-Asym, Sack, Fack, Vegas and Vegas-RBP as source and five - TCPSink, TCPSink-Asym, Sack, DelAck and Sack1-DelAck as destination, implemented in Network Simulator (NS-2). Performance of TCP versions indicates how they respond to various network parameters-propagation delay, bandwidth, TTL (time to live), RTT (round trip time), rate of packet sending and so on. Such analysis is immensely in need to be aware of which TCP is better for a specific criterion, wherefrom an appropriate one will be selected in respective network to optimize traffic goal.

P. Omprakash et al. [6], focused on how TCP will be serviced by WiMAX, and what are the issues that are still open and can be used to increase the performance of the service. First it was reviewed the throughput of TCP tahoe against time. Then, the TCP variants were compared by respective throughput against time. Some of the flavors of TCP congestion control are loss-based, high-speed TCP congestion control algorithms that uses packet losses as an indication of congestion; delay-based TCP congestion control that emphasizes packet delay rather than packet loss as a signal to determine the rate at which to send packets. The authors compared three TCP variants, namely Tahoe, New Reno and Vegas were On the basis of throughput, round-trip time (RTT) and packet loss ratio. While all the TCP variants achieve similar throughput, they do so in different ways, with different impacts on the network performance. The adverse effects of TCP window auto-tuning is identified in this environment and demonstrate that on the downlink, congestion losses dominate wireless transmission error. Several issues were revealing for this WiMAX-based networks, including limited bandwidth for TCP, high RTT and jitter, and unfairness during remote login, VoIP, and video streaming.

F. Furqan Doan et al. [7], propose a mechanism namely WiMAX Fair Intelligent Congestion Control (WFICC) to avoid congestion at the base station. WFICC ensures that the traffic is scheduled in such a way that the base station output buffer operates at a target operating point, without violating the QoS requirements of connections. A detailed simulation study is performed in ns-2 to evaluate the effectiveness of proposed algorithm to meet the QoS requirements of different Class of Services (CoSs). The results have shown that the proposed WFICC algorithm enables the base station to avoid congestion and ensures the provision of QoS of different Class of Services (CoSs) in terms of throughput, fairness and packet delay.

III. PROPOSED METHODOLOGY

A lot of research has been made so far, aiming at suggesting ways to improve the efficiency of TCP in wireless networks. However, there are not extensive comparative studies of TCP performance in WiMAX networks. In this work, we evaluate some representative TCP congestion control schemes under various traffic scenarios, which include single and multiple TCP flows through WiMAX networks in the presence of wireless channel errors, network asymmetries and various level of link congestion. The target is to find out the best performing TCP schemes and to suggest ways for further improvements..

A. Simulation And Results

In this work, main aim is to simulate and analyze performance of three TCP-variants under varying mobility rate over WiMAX environment. Simulations are done considering a network of 50 mobile nodes placed randomly within 1600×1600 m^2 area. File Transfer Protocol (FTP) data sessions among randomly chosen 25 subscriber nodes with base station are used. However, during this data transfer process, all nodes will operate in the background for providing the necessary support to the ongoing communication process in the network. Each simulation is executed for 200 seconds. The subscriber nodes are considered to be mobile and simulations are performed by varying their mean speed.

A. **Network Simulator – ns2**

Network Simulator (Version 2), widely known as NS2, is simply an event driven simulation tool that has proved useful in studying the dynamic nature of communication networks. Simulation of wired as well as wireless network functions and protocols (e.g., routing algorithms, TCP, UDP) can be done using NS2. In general, NS2 provides users with a way of specifying such network protocols and simulating their corresponding behaviours. Due to its flexibility and modular nature, NS2 has gained constant popularity in the networking research community since its birth in 1989. Ever since, several revolutions and revisions have marked the growing maturity of the tool. Among these are the University of California and Cornell University who developed the REAL network simulator, the foundation which NS is based on. Since 1995 the Defence Advanced Research Projects agency (DARPA) supported development of NS through the Virtual Inter Network Test-bed (VINT) project. Currently the National Science Foundation (NSF) has joined the ride in development. Last but not the least, the group of researchers and developers in the community are constantly working to keep NS2 strong and versatile. Fig. gives an overview of C++ class hierarchies. The entire hierarchy consists of over 100 C++ classes and struct data types.

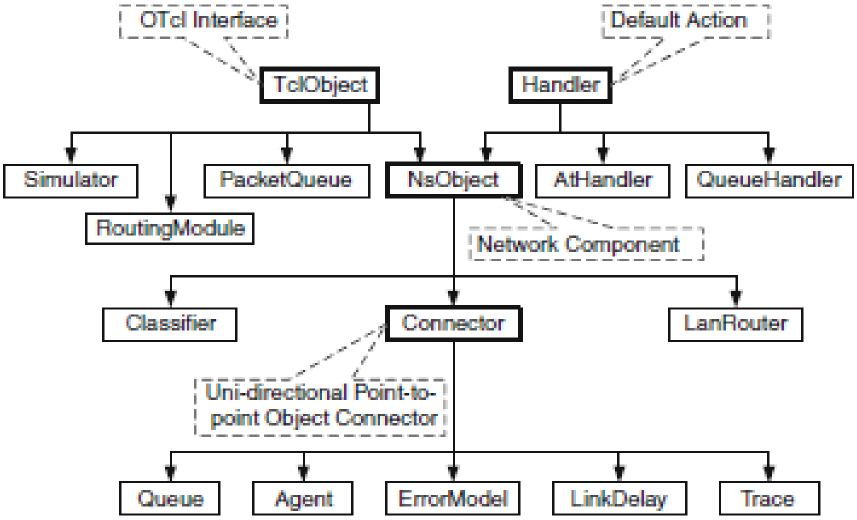

FIGURE 5:. CLASS Hierarchy.

All classes deriving from class TclObject form the compiled hierarchy. Classes in this hierarchy can be accessed from the OTcl domain. For example, they can be created by the global OTcl procedure “new{...}”. Classes derived directly from class TclObject include network classes (e.g., NsObject), packet-related classes (e.g., PacketQueue),Simulation-related classes (e.g., Scheduler), and helper classes (e.g., Routing- Module). Again, classes which do not need OTcl counterparts (e.g., classes derived from class Handler) form their own standalone hierarchies. These hierarchies are not a part of the compiled hierarchy nor the interpreted hierarchy.

Table I *Salient Simulation Parameters*

Parameter	**Value**
Simulation time	200 Sec
Simulation area	1600m x 1600m
Quality of Service	Omni antenna
No. of nodes	Best Effort
TCP Varients	TCP-Tahoe, Reno, Westwood
Modulation & Coding	BPSK 1/2
Traffic	FTP
TCP segment size	1024 bytes
Cyclic prefix	1/4
Frame duration	10 msec
Mean speed	30, 40, 50, 60, 70, 80

.

C. Results and Analysis

In this work, we have used four performance matrics namely throughput, average end to end delay, packet delivery ratio and routing overhead, to evaluate and analyse the performance of TCP-newreno, westwood and cubic under different mean speed scenarios.

1. **Throughput**

Throughput is the ratio of total number of delivered or received data packets to the total duration of simulation time.Mathematically,

$$Throughput = \frac{\sum_{1}^{n} CBRrece}{simulationtime}$$

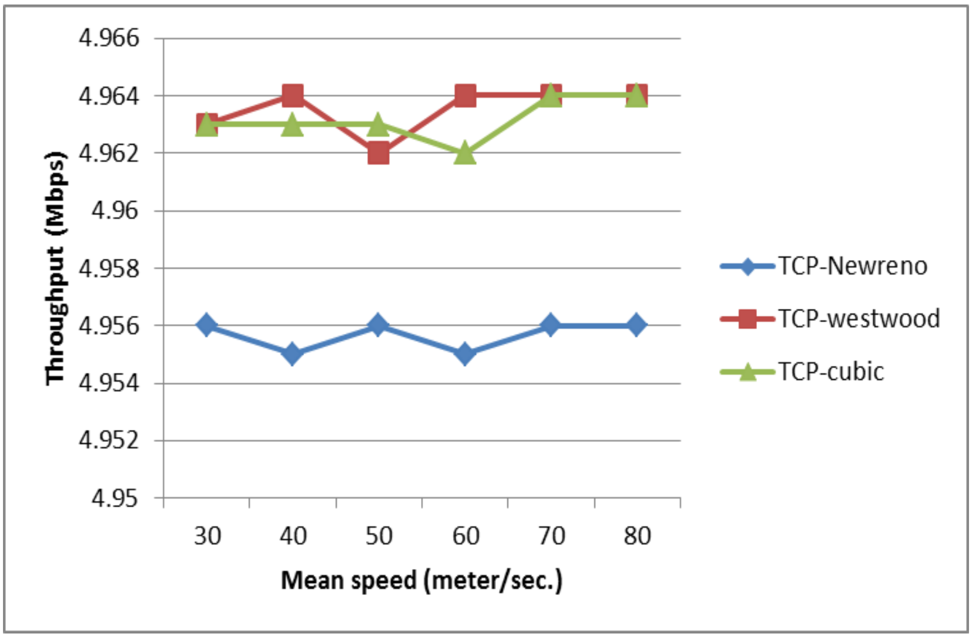

Figure 6: Throughput versus mean speed.

2. **Packet Delivery Ratio (PDR)**

PDR also known as the ratio of the data packets delivered to the destinations to those generated by the CBR sources. This metric characterizes both the completeness and correctness of the routing protocol.

$$PDR = \frac{\sum_{1}^{n} CBRrece}{\sum_{1}^{n} CBRsent} * 100$$

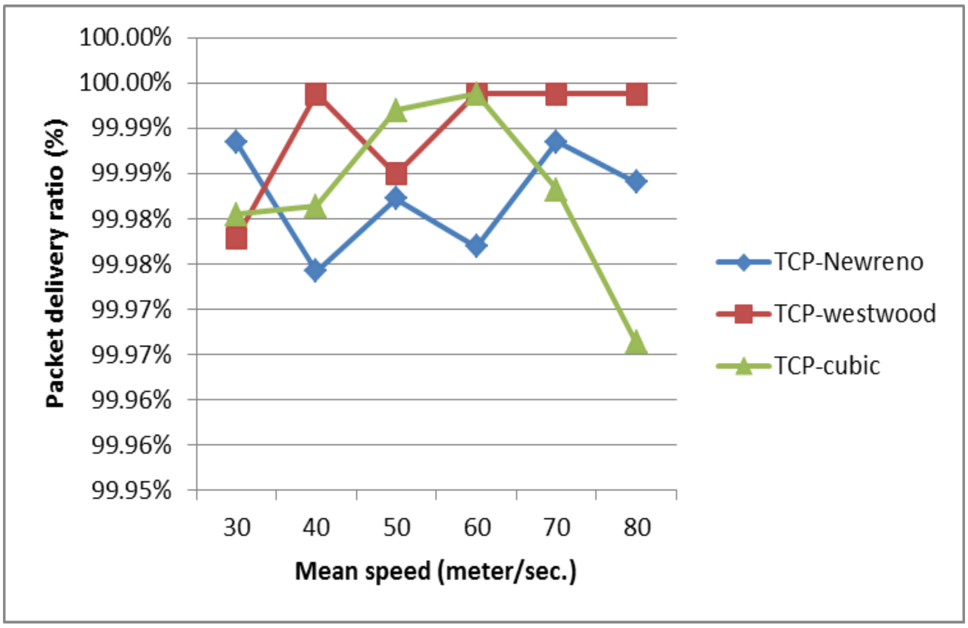

Figure 7: PDR versus mean speed.

3. **Average End to End Delay**

Average End to End delay is the average time taken by a data packet to reach from source node to destination node. It is ratio of total delay to the number of packets received. Mathematically,

$$Avg\_End\_to\_End\_Delay = \frac{\sum_{1}^{n}(CBRrecetime - CBRsenttime)}{\sum_{1}^{n} CBRrece} * 100$$

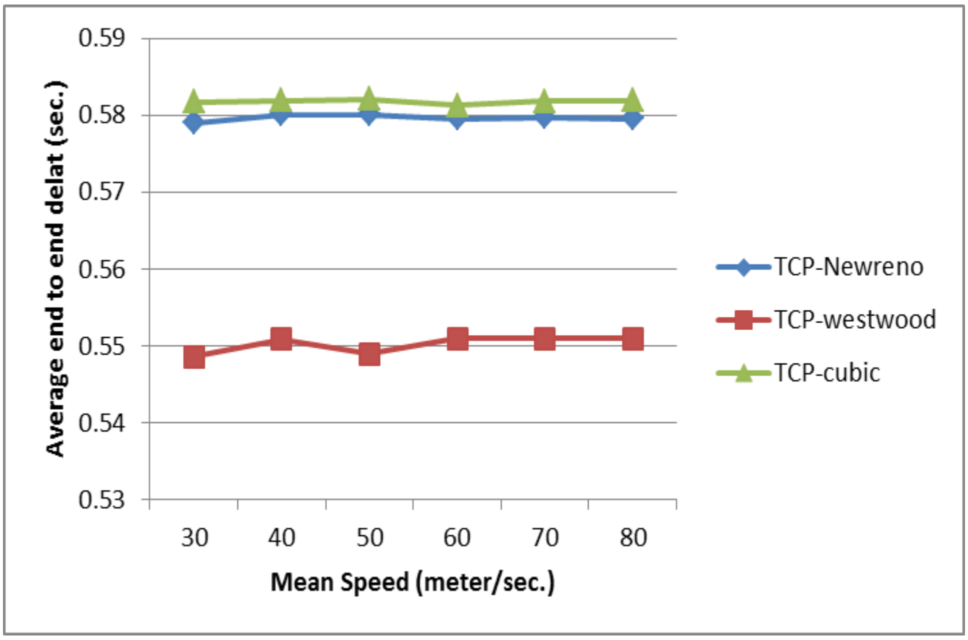

Figure 8: Average delay versus mean speed.

4. Normalized Protocol Overhead/ Routing Load

Routing Load is the ratio of total number of the routing packets to the total number of received data packets at destination.

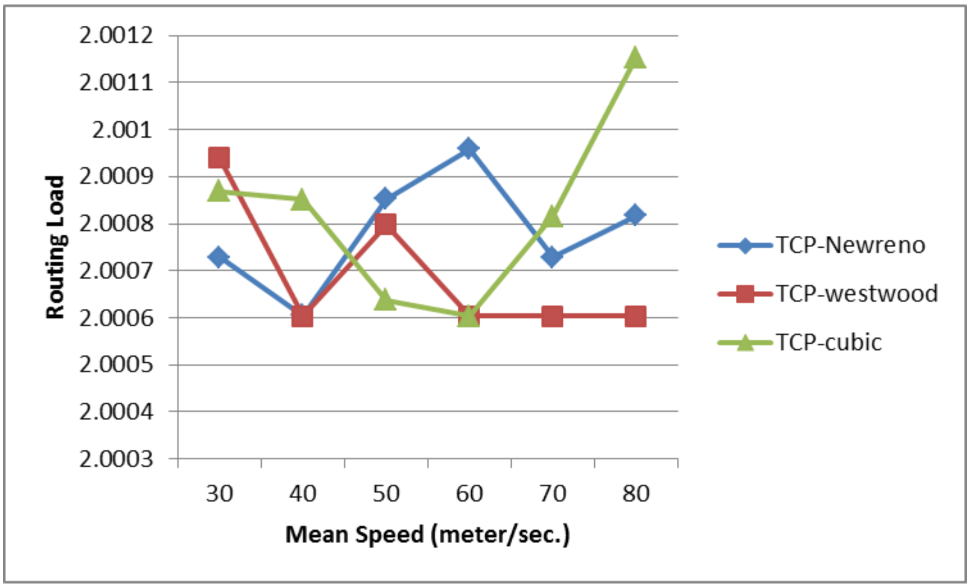

Figure 9: Routing load versus mean speed.

From the results obtained, we can conclude that TCP cubic has a higher performance than other TCP variants. The real-time nature of TCP-Cubic keeps the window growth rate independent of RTT, which keeps the protocol TCP friendly under both short and long RTT paths.

IV. CONCLUSION AND FUTURE WORK

In this paper, we have analyzed the performance of the important TCP variants like TCP New Reno, TCP Westwood, and TCP Cubic in WiMAX environment. Through simulation, we noted that TCP throughput decreases significantly when node movement cases link failures, due to TCP's inability to recognize the difference between link failure and congestion. from the view of throughput, average delay and packet delivery ratio, TCP cubic is the best congestion control scheme out of selected TCP variants. However, from the view of average delay, TCP wesstwood shows better results than TCP cubic. From this analysis, we found that TCP cubic is better than other TCP variants in case of increasing Random Packet Loss as well as in case of increasing mobility.

On the basis of the results obtained from simulation graphs and some trials in the literature, we can extend this work by improving TCP-cubic performance over WiMAX wnvironment using QoS bandwidth allocation schemes.

.**References**

[1] B. Chen, I. Marsic, R. Miller, "*Issues and Improvements in TCP Performance over Multihop Wireless Networks*", IEEE Sarnoff Symposium, 2008.

[2] Georgi Kirov, "*A Simulation Analysis of the TCP Control Algorithms*", International Conference on Computer Systems and Technologies, *2005*.

[3] Konstantinos Tsiknas, George Stamatelos," *Performance Evaluation of TCP in IEEE 802.16 Networks",* Wireless communication and networking conference : mobile and wireless network, IEEE, 2012.

[4] M. Gerla, K. Tang and R. Bagrodia, "*TCP performance in wireless multi-hop networks*", Proceedings of WMCSA, IEEE, February 1999.

[5] Kartik Chandran, Sudarshan Raghunathan, S. Venkatesan, Ravi Prakash ," A Feedback Based Scheme For Improving TCP Performance In Ad-Hoc Wireless Networks", Personal Communications, IEEE, vol. 8,pp. 34 – 39, 2001.

[6] Md. Shohidul Islam, M.A Kashem, W.H Sadid, M. A Rahman, M.N Islam," *TCP Variants and Network Parameters: A Comprehensive Performance Analysis*", Proceedings of the International Multi-Conference of Engineers and Computer Scientists, vol. 2, pp - 978-988, 2009.

[7] P. Omprakash, R. Sabitha," *Performance Analysis of TCP over WiMAX",* IEEE, 2011.

[8] Fatima Furqan and Doan B. Hoang, " WFICC: A New Mechanism for Provision of QoS and Congestion Control in WiMAX", The 10th Annual IEEE CCNC- Wireless Networking, IEEE, 2013.

[9] T. Anouar and A. Haqiq ," *Performance Analysis of VoIP Traffic in WiMAX using various Service Classes*" International Journal of Computer Applications, vol. -20, 2012, pp. 29-33.

[10] Shraddha Bansal, Raksha Upadhyay, "*Performance Improvement of Wi-Max IEEE 802.16e in Presence of Different FEC Codes*", CICSYN, IEEE, 2009.

[11] M. Rehan Rasheed ,*" Performance of Routing Protocols in WiMAX Networks",* International Journal of Engineering and Technology, vol.2, no.5, October 2010.

[12] Injong Rhee, and Lisong Xu ," CUBIC: A New TCP-Friendly High-Speed TCP Variant", SIGOPS, ACM, vol.- 42,2008.

