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

A Review on Quality of Service Improvement in WiMax

Manju*, Lovnish Bansal
Dept. of Computer Science & Engineering
B.M.S.C.E, Mukatsar, India

Abstract: The evergreen demand for fast delivery of large volumes of data is one of the challenging task for wireless communication technology. WiMAX (Worldwide Interoperability for Microwave Access) is a wireless broadband solution that offers a rich set of features with a lot of flexibility in terms of deployment options and potential service offerings. Its main objective is to provide quality with cost effectiveness. But Delivering QoS is more challenging for mobile broadband than for fixed. The time variability and unpredictability of the channel become more acute, and complication arises from the need to hand over sessions from one cell to another as the user moves across their coverage boundaries. We discussed various work on congestion avoidance in WiMAX, proposed in the WiMAX research literature. This work presents a survey that WIMAX is the next generation broadband wireless technology which offers high speed, secure, sophisticate and last mile broadband services. However it also poses some congestion related problems that uses packet losses as an indication of congestion. We concentrate on two flavors of tcp that is tcp sack and tcp fack that are used as a congestion avoidance algorithms to improve the Average End-to-End Delay, Packet Delivery Ratio, Packets Lost, Routing Overhead of the connection. Thus it will improve the QoS in Wimax.

Keywords: Wimax, congestion avoidance, tcp sack, tcp fack, QoS.

I. INTRODUCTION

WiMAX (Worldwide Interoperability for Microwave Access) is a wireless communications standard designed to provide 30 to 40 megabit-per-second data rates, with the 2011 update providing up to 1 Gbit/s for fixed stations. It is an emerging broadband radio access technology, has attracted much attention from the whole telecom industry in recent years. Its main features are high-speed transmission rate, large coverage, support for mobility, QoS guarantee and all-IP architecture. The technology fulfills the integration of packetized data, broadband access and mobilized terminal; therefore, it has a bright future for wide application. Its name "WIMAX" was created by the WIMAX Forum, which was formed in June 2001 to promote conformity and interoperability of the standard. The forum describes WIMAX as "a standards-based technology enabling the delivery of last mile wireless broadband access as an alternative to cable and DSL". WIMAX refers to interoperable implementations of the IEEE 802.16 family of wireless-networks standards ratified by the WIMAX Forum. WIMAX Forum certification allows vendors to sell fixed or mobile products as WIMAX certified, thus ensuring a level of interoperability with other certified products, as long as they fit the same profile. The bandwidth and range of WiMAX Provides portable mobile broadband connectivity across cities and countries through a variety of devices, it also provides a wireless alternative to cable and digital subscriber line (DSL) for "last mile" broadband access. Inspite of all these tremendous facilities QoS remains one major factor that most of the wireless technologies seeks as a challenge. We surveyed various papers presented in this regard and come up with two algorithms of improving the quality of service (QoS) in the WiMAX network. These two algorithms are basically variants of TCP i.e tcp-sack and tcp-fack forming the congestion avoidance algorithms. Section 2 gives the background on IEEE 802.16 and wimax. Section 3 covers the overview of literature survey that serves as the foundation approaches to the algorithms we surveyed. Lastly, Conclusions and future scope comes up in section 5.

II. BACKGROUND

The original IEEE 802.16 standard (now called "Fixed WIMAX") was published in 2001. WIMAX adopted some of its technology from WiBro, a service marketed in Korea. Mobile WIMAX (originally based on 802.16e-2005) is the revision that was deployed in many countries, and basis of future revisions such as 802.16m-2011.WIMAX is sometimes referred to as "Wi-Fi. on steroids" and can be used for a number of applications including broadband connections. It is similar to Wi-Fi, but it can enable usage at much greater distances. WIMAX is called the next generation broadband wireless technology which offers high speed, secure, sophisticate and last mile broadband services along with a cellular back haul and Wi-Fi hotspots. The evolution of WIMAX began a few years ago when scientists and engineers felt the need of having a wireless Internet access and other broadband services which works well everywhere especially the rural areas or in those areas where it is hard to establish wired infrastructure and economically not feasible. IEEE 802.16, also known as IEEE Wireless-MAN, explored both licensed and unlicensed band of 2-66 GHz which is standard of fixed wireless broadband and included mobile broadband application. WIMAX forum, a private organization was formed in June 2001 to coordinate the components and develop the equipment those will be compatible and inter operable. After several years, in 2007, Mobile WIMAX equipment developed with the standard IEEE 802.16e got the certification and they announced to release the product in 2008, providing mobility and nomadic access. The IEEE 802.16e air interface based on Orthogonal Frequency Division Multiple Access (OFDMA) which main aim is to give better performance in non-line-of-sight environments. IEEE 802.16e introduced scalable channel bandwidth up to 20 MHz, Multiple Input Multiple Output (MIMO) and AMC enabled 802.16e technology to support peak Downlink (DL) data rates up to 63 Mbps in a 20 MHz channel through Scalable OFDMA (S-OFDMA) system.

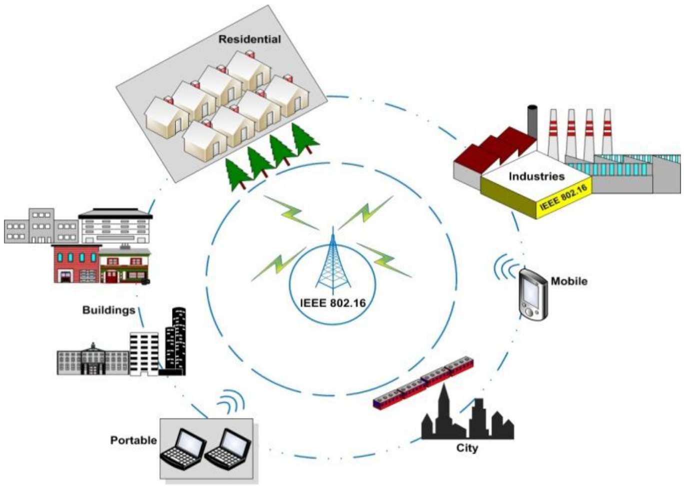

Fig. 1 IEEE 802.16 / WiMAX network architecture

III. LITERATURE SURVEY OF VARIOUS CONGESTION CONTROL ALGORITHMS

A. Tcp Tahoe

TCP Tahoe was the first algorithm to employ two transmission phases: slow start and congestion avoidance. TCP is based on a principle of 'conservation of packets', i.e. if the connection is running at the available bandwidth capacity then a packet is not injected into the network unless a packet is taken out as well. TCP implements this principle by using the acknowledgements to clock outgoing packets because an acknowledgement means that a packet was taken off the wire by the receiver. It also maintains a congestion window CWD to reflect the network capacity.

1. *Slow Start:* To resolve the congestion problem the slow start algorithm has been involved. The basic of this approach is the notion of a congestion window (cwnd). When the new connection is established the cwnd is initialized to one packet. Every time a packet with sequence number n arrives at the receiver, the receiver confirms the packet n by sending an acknowledgement (ACK) packet. It contains the information for the sequence number of another packet, which it is waiting for (may not be "n+1") back to the sender. TCP uses an arrival of ACK as a trigger of new packet transmission, i.e. each time an ACK is received, the congestion window is increased by one packet. The sender stops increasing the window size when one reach the limit of the network capacity. The limit is defined as minimum of window that sender can transmit and window that receiver can receive.
2. *Congestion Avoidance:* For congestion avoidance Tahoe uses 'Additive Increase Multiplicative Decrease'. A packet loss is taken as a sign of congestion and Tahoe saves the half of the current window as a threshold value. It then set CWD to one and starts **slow start** phase until it reaches the threshold value. After that it increments linearly until it encounters a packet loss. Thus it increases its window slowly as it approaches the bandwidth capacity.
3. *Problems:* The problem with Tahoe is that it takes a complete timeout interval to detect a packet loss and in fact, in most implementations it takes even longer because of the coarse grain timeout. Also since it doesn't send immediate ACK's, it sends cumulative acknowledgements, therefore it follows a 'go back n' approach. Thus every time a packet is lost it waits for a timeout and the pipeline is emptied. This offers a major cost in high band-width delay product links.

B. Tcp Reno

The Reno retains the basic principle of Tahoe, such as slow starts and the coarse grain re-transmit timer. However it adds some intelligence over it so that lost packets are detected earlier and the pipeline is not emptied every time a packet is lost. Reno requires that we receive immediate acknowledgement whenever a segment is received. The logic behind this is that whenever we receive a duplicate acknowledgment, then his duplicate acknowledgment could have been received if the next segment in sequence expected, has been delayed in the network and the segments reached there out of order or else that the packet is lost. If we receive a number of duplicate acknowledgements then that means that sufficient time has passed and even if the segment had taken a longer path, it should have gotten to the receiver by now. There is a very high probability that it was lost. So Reno suggests an algorithm called **'Fast Re-Transmit'.** Whenever we receive 3 duplicate ACK's we take it as a sign that the segment was lost, so we re-transmit the segment without waiting for timeout. Thus we manage to re-transmit the segment with the pipe almost full. Another modification that RENO makes is in that after a packet loss, it does not reduce the congestion window to 1, since this empties the pipe. It enters into a algorithm which we call **'Fast-Re-Transmit'.** The basic algorithm is presented as under:

1: Each time we receive 3 duplicate ACK's we take that to mean that the segment was lost and we re-transmit the segment immediately and enter 'Fast-Recovery'.
2: Set SSthresh to half the current window size and also set CWD to the same value.
3: For each duplicate ACK receive increase CWD by one. If the increase in CWD is greater than the amount of data in the pipe then transmit a new segment else wait. If there are 'w' segments in the window and one is lost, then we will receive (w-1) duplicate ACK's. Since CWD is reduced to W/2, therefore half a window of data is acknowledged before we can send a new segment. Once we retransmit a segment, we would have to wait for atleast one RTT before we would receive a fresh acknowledgement. Whenever we receive a fresh ACK we reduce the CWND to SSthresh. If we had previously received (w-1) duplicate ACK's then at this point we should have exactly w/2 segments in the pipe which is equal to what we set the CWND to be at the end of fast recovery. Thus we don't empty the pipe, we just reduce the flow. We continue with congestion avoidance phase of Tahoe after that.

1. *Problems:* Reno performs very well over TCP when the packet losses are small. But when we have multiple packet losses in one window then RENO doesn't perform too well and its performance is almost the same as Tahoe under conditions of high packet loss.

C. Tcp New-Reno

New RENO is a slight modification over TCP-RENO. It is able to detect 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 doesn't exit fast-recovery until all the data which was out standing at the time it entered fast recovery is acknowledged. Thus it overcomes the problem faced by Reno of reducing the CWD multiples times. The fast-transmit phase is the same as in Reno. The difference in the fast recovery phase which allows for multiple re-transmissions in new-Reno. Whenever new-Reno enters fast recovery it notes the maximums segment which is outstanding. The fast-recovery phase proceeds as in Reno, however when a fresh ACK is received then there are two cases: If it ACK's all the segments which were outstanding when we entered fast recovery then it exits fast recovery and sets CWD to ssthresh and continues congestion avoidance like Tahoe. If the ACK is a partial ACK then it deduces that the next segment in line was lost and it re-transmits that segment and sets the number of duplicate ACKS received to zero. It exits Fast recovery when all the data in the window is acknowledged**.**

1. *Problems:* New-Reno suffers from the fact that it take one RTT to detect each packet loss. When the ACK for the first retransmitted segment is received only then can we deduce which other segment was lost.

D. Tcp Vegas

TCP Vegas is a TCP congestion control algorithm that emphasizes packet delay, rather than packet loss, as a signal to determine the rate at which to send packets. TCP Vegas detects congestion based on increasing Round Trip Time (RTT) values of the packets in the connection unlike TCP Reno which detect congestion only after it has actually happened via packet drops.
The algorithm depends heavily on accurate calculation of the Base RTT value. Base RTT is set to be the minimum of all measured RTTs; it is commonly the RTT of the first segment sent by the connection. If the connection is not over flown by the traffic, the expected throughput is given by:
Expected Throughput = WindowSize/BaseRTT
Where WindowSize is the size of the current congestion window
Then current actual sending rate is calculated once per round trip time as:
Actual throughput = W/RTT
Where W is the congestion window
The congestion window is adjusted depending upon the difference between expected and actual sending rates.
Difference = (expected - actual) baseRTT
Also two thresholds a and b are defined such that, $a < b$ and $a > b$ correspond to having too little and too much extra traffic in the network, respectively. When Difference < a, TCP Vegas increases the congestion window linearly during the next RTT, and when Difference > b, TCP Vegas decrease the congestion window linearly during the next RTT. The congestion window is
Left unchanged when a < Difference < b.

1. *Problems:* This algorithm emphasizes on packet delay rather than packet loss as a signal to determine the rate at which to send packets. It also depends heavily on accurate calculation of the Base RTT value. TCP Vegas detects congestion based on increasing Round Trip Time (RTT) values of the packets in the connection unlike TCP Reno which detect congestion only after it has actually happened via packet drops.

E. Tcp Veno

TCP Veno operates with the objective of distinguishing loss type, thus performing an appropriate window reduction instead of a fixed drop of window in Reno, and forcing a TCP connection to stay longer at the equilibrium by employing a proactive congestion detection method and a reactive congestion detection method together. Veno differs from the conventional TCP in two ways: 1) It dynamically adjusts the slow start threshold *(ssthresh)* based on the equilibrium estimation of a connection as opposed to using a fixed drop-factor window,when packet loss is encountered. 2) It uses a refined linear increase algorithm, which employs both the proactive and reactive congestion detection schemes to adjust the congestion window size during the additive increase phase.

1. *Problems:* TCP Veno give same problems as Reno in performing data retransmission when multiple packets are lost in a single of window.

F. Tcp Sack

TCP with 'Selective Acknowledgments' 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 retransmit parts of RENO. SACK TCP requires that segments not be acknowledged cumulatively but should be acknowledged selectively. Thus each ACK has a block which describes which segments are being acknowledged. Thus the sender has a picture of which segments have been acknowledged and which are still outstanding. Whenever the sender enters fast recovery, it initializes a variable pipe which is an estimate of how much data is outstanding in the network, and it also set CWND to half the current size. Every time it receives an ACK it reduces the pipe by 1 and every time it retransmits a segment it increments it by 1. Whenever the pipe goes smaller than the CWD window it checks which segments are unreceived and send them. 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.

G. Tcp Fack

The FACK algorithm uses the additional information provided by the SACK option to keep an explicit measure of the total number of bytes of data outstanding in the network. In doing so,fack attempts to preserve TCP's self clock and reduce the overall burstiness of TCP. The requisite network state information can be obtained with accurate knowledge about the forward most data held by the receiver, by forward most, we mean the correctly received data with the highest sequence number. Also the sender must retain information on data blocks held by the receiver, which is required in order to use SACK information to correctly retransmit data. In addition to what is needed to control data retransmission information on retransmitted segments must be kept in order to accurately determine when they have left the network.

IV. CONCLUSION AND FUTURE SCOPE

Among these conclusions some algorithms varies from one scheme to the other. In general most of the schemes lack with practical implementation. Moreover, those who have been implemented are limited to a particular environment. Lack of studies about these schemes is also an issue. It is a well known fact that WiMax suffers with different issues. Some of the most prominent issues are bandwidth constraints and limited power of mobile devices and some other are packet's random losses and retransmission timeouts(with given load) in high delay networks that leads to congestion. So in future work, TCP sender side mechanisms to handle these higher offered load, random losses and retransmission timeouts in high delay networks in such a way as to keep congestion window as high as possible, while keeping the congestion under control and keep retransmissions to minimal. Most of the schemes mentioned above clearly lacks in handling this and some other issues. Therefore there is definitely a need of congestion avoidance algorithms that can offer a better QoS improvement in WiMAX. From the above discussion TCP SACK and TCP FACK delivers effective throughput and packet delivery ratio. Here the TCP proposed mechanisms are assessed against TCP SACK and TCP FACK mechanisms to see how they fair against congestion and high offered load in the network.

REFERENCES

[1] B. Chen, I. Marsic, and R. Miller, "Issues and Improvements in TCP Performance over Multihop Wireless Networks", IEEE Sarnoff Symposium, pp. 1-5, 2008.

[2] C. Chen, H. Wang, XinWang, M. Li, A.O. Lim, "A Novel Receiver-aided Scheme for Improv-ing TCP Performance in Multihop Wireless Networks", in Proc. of Int. Conference on Communications and Mobile Computing, pp. 272–277, 2009.

[3] Ruhani Ab Rahman ,"Performance Analysis of Routing Protocol in WiMAX Network", IEEE International Conference on System Engineering and Technology,2011.

[4] Tarik Anouari ,"Performance Analysis of VoIP Traffic in WiMAX using various Service Classes", International Journal of Computer Applications ,Volume 52– No.20, pp. 29-34, August 2012.

[5] Vikram Mehta ,"Performance Analysis of QoS Parameters for Wimax Networks", International Journal of Engineering and Innovative Technology, Volume 1, Issue 5, pp. 105-110, May 2012.

[6] M. Rehan Rasheed ," Performance of Routing Protocols in WiMAX Networks", International Journal of Engineering and Technology, Vol.2, No.5, October 2010.

[7] Jeng-Ji Huang and Jeng-Ji Huang, "A New Method to Improve the Performance of TCP SACK over Wireless Links", the National Science Council, Taiwan (NSC 91-2219-E-260-001),volume 3, pp. 1730-1734, 2003.

[8] Chih-He Chiang, Wanjiun Liao, Tehuang Liu, Iam Kin Chan, and Hsi-Lu Chao, "Adaptive Downlink and Uplink Channel Split Ratio Determination for TCP-Based Best Effort Traffic in TDD-Based WiMAX Networks", IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, VOL. 27, NO. 2, pp. 182-190, FEBRUARY 2009.

[9] Md. Mahbubul Alam, ID- 09221116 And Tanmoon Taz Shetu, ID- 09221108," Congestion control in Mobile Ad-Hoc Networks (MANETs)", Department of Computer Science and Engineering,April 2011.

[10] Matthew Mathis and Jamshid Mahdavi," Forward Acknowledgment_ Refining TCP Congestion Control", National Science Foundation volume 26 number 4, pp. 1-11, October 1996.

[11] Chung Ling Chi, Fu Chengpeng, and Liew Soung Chang, "Improvements Achieved by SACK Employing TCP Veno Equilibrium-Oriented Mechanism over Lossy Networks", EUROCON'2001,Trends in Communications, International Conference on. ,Volume 1, pp. 202-209, 2001.

[12] Jin Hwang, Sang Woo Son, Byung Ho Rhee," Improving TCP Performance over WiMAX Networks Using Cross-layer Design",Convergence and Hybrid Information Technology, 2008. ICCIT '08. Third International Conference on ,Volume 2, pp. 83-88, 2008.

[13] Rakesh Kumar Jha, Idris Z. Bholebawa, Upena D. Dalal," Location Based Performance of WiMAX Network for QoS with Optimal Base Stations (BS)", Wireless Engineering and Technology, Volume 2, No.3, pp. 136-145, July 2011.

[14] P Omprakash ,R Sabitha ," PERFORMANCE ANALYSIS OF TCP OVER WiMAX", Electronics Computer Technology (ICECT), 2011 3rd International Conference on ,Volume 6, pp. 348-352, 2011.

[15] L.D.Malviya A. Gaiwak Dr. P.D.Vyavhare," Simulation based comparison of different Modulation schemes for Mobile WiMAhhjhjmnnX using TCP and its Variants", 978-0-7695-3267-7/08 $25.00 © 2008 IEEE DOI 10.1109/ICETET.2008.191, pp. 168-171, First International Conference on Emerging Trends in Engineering and Technology.

