An Improved TCP Westwood Congestion Avoidance Algorithm To Increase Throughput

Vivekkumarmajithiya[1] Prof. Priyang Bhatt[2]
[1]Department Of Information Technology [2]Department Of Computer Engineering
G.H.Patel College Of Engineering & Technology Vallabhvidyanagar,Gujarat, India

Abstract—TCP is a transport layer protocol which provides reliable and connection oriented transmission. The main problem of TCP is in wireless networks. In wired network, packet loss is mostly due to network congestion, while in wireless network the packet loss may be due to network congestion or may be due to mobility of nodes. TCP starts its back off procedure when it detects packet loss, if the packet loss is due to node movement then the back off procedure will be unnecessary and it will create delay in the network. There are many TCP variants available. TCP Westwood (TCPW) is a sender-side-only modification to TCP New Reno that is intended to better handle large bandwidth-delay product paths (large pipes), with potential packet loss due to transmission or other errors (leaky pipes), and with dynamic load (dynamic pipes). In this paper a better mechanism for congestion avoidance is suggested for TCP Westwood to increase throughput.

Keywords: MANET, TCP, back off procedure

I. INTRODUCTION

Mobile ad hoc network is an autonomous and self-configuring network [1] composed of several mobile nodes i.e. mobile equipments like cell phones, PDAs, tablets etc. These mobile nodes communicate with each other through the wireless links established between nodes. Furthermore the communications between nodes occur without the support of any infrastructure or any centralized access point like a Base Station. Thus MANET never forms any fixed topology as it is dynamic in nature.

The Transmission Control Protocol (TCP) [2] is used bythe vast majority of Internet applications. The TCP is a connection-oriented transport protocol that provides a reliablebyte-stream data transfer service between pairs ofprocesses.When two processes want to communicate, they mustfirst establish a TCP connection (initialize the status informationon each side). Since connections must be establishedbetween unreliable hosts and over the unreliableInternet communication system, a handshake mechanism with clock based sequence numbers is used.The TCP provides multiplexing facilities by usingsource and destination port numbers. These port numbers allow the TCP to set up virtual connections over a physicalconnection and multiplex the data streams through thatconnection.Each connection is uniquely specified by a pair of sockets1identifying its two sides and by specific parameterssuch as sequence numbers, window sizes, and status information(necessary for reliability and flow and congestioncontrol mechanisms – see the following for more details).At the end of a communication, the connection is terminatedor closed to free the resources for other uses.The TCP is able to transfer a continuous stream of bytes, in each direction, by packaging some number of bytes into segments (TCP data unit). The size of these segments and the timing at which they are sent is generally left to the TCP module. However, an application can force delivery of segments to the output stream using a push operation provided by the TCP to the application layer. A push causes theTCP to promptly forward and deliver data to the receiver.Apart from this basic data transfer, the TCP provides some more services. First of it is able to recover from datathat are damaged, lost, duplicated, or delivered out of orderby the Internet communication system (reliability).Missing or corrupted segments are detected andretransmitted assigning a sequence number to each transmittedsegment, and requiring a positive acknowledgment(ACK) from the receiver. If the ACK is not received within atimeout interval, the data are retransmitted.At the receiver, the sequence numbers are used to correctlyorder segments that may be received out of order and to eliminate duplicates. Damage is handled by adding a checksum to each transmitted segment, checking it at thereceiver, and discarding damaged segments.

The TCP also provides a means for the receiver to governthe amount of data sent by the sender (flow control). Thisis achieved by returning a ‘‘window’’, named advertisedwindow, that indicates the allowed number of bytes that the sender may transmit before receiving furtherpermission.

Moreover, TCP users may indicate the security and precedence of their communication. Provision is made for defaultvalues to be used when these features are not needed. And the most important feature provided bt TCP is TCP congestion control

II. TCPCONGESTION CONTROL ALGORITHMS

The four algorithms, Slow Start, Congestion Avoidance,Fast Retransmit and Fast Recovery [3][4] are describedbelow.

A. Slow start

Slow Start, a requirement for TCP softwareimplementations is a mechanism used by the sender tocontrol the transmission rate, otherwise known as senderbasedflow control. This is accomplished through the returnrate of acknowledgements from the receiver. In other words,the rate of acknowledgements returned by the receiverdetermines the rate at which the sender can transmit data.

When a TCP connection first begins, the Slow Startalgorithm initializes a congestion window to one segment,which is the maximum segment size (MSS) initialized bythe receiver during the connection establishment phase.When acknowledgements are returned by the receiver, thecongestion window increases by one segment for eachacknowledgement returned. Thus, the sender can transmitthe minimum of the congestion window and the advertisedwindow of the receiver, which is simply called thetransmission window.At some point the congestion

window may become toolarge for the network or network conditions may changesuch that packets may be dropped. Packets lost will trigger atimeout at the sender. When this happens, the sender goesinto congestion avoidance mode as described in the nextsection.

B. Congestion avoidance

During the initial data transfer phase of a TCPconnection the Slow Start algorithm is used. However, theremay be a point during Slow Start that the network is forcedto drop one or more packets due to overload or congestion.If this happens, Congestion Avoidance is used to slow thetransmission rate. However, Slow Start is used inconjunction with Congestion Avoidance as the means to getthe data transfer going again so it doesn't slow down andstay slow.

In the Congestion Avoidance algorithm a retransmissiontimer expiring or the reception of duplicate ACKs canimplicitly signal the sender that a network congestionsituation is occurring. The sender immediately sets itstransmission window to one half of the current window size(the minimum of the congestion window and the receiver'sadvertised window size), but to at least two segments. Ifcongestion was indicated by a timeout, the congestionwindow is reset to one segment, which automatically putsthe sender into Slow Start mode. If congestion was indicatedby duplicate ACKs, the Fast Retransmit and Fast Recoveryalgorithms are invoked.

As data is received during Congestion Avoidance, thecongestion window is increased. However, Slow Start isonly used up to the halfway point where congestionoriginally occurred. This halfway point was recorded earlieras the new transmission window. After this halfway point,the congestion window is increased by one segment for allsegments in the transmission window that areacknowledged. This mechanism will force the sender tomore slowly grow its transmission rate, as it will approachthe point where congestion had previously been detected.

C. Fast retransmit

When a duplicate ACK is received, the sender does notknow if it is because a TCP segment was lost or simply thata segment was delayed and received out of order at thereceiver. If the receiver can re-order segments, it should notbe long before the receiver sends the latest expectedacknowledgement. Typically no more than one or twoduplicate ACKs should be received when simple out oforder conditions exist. If however more than two duplicateACKs are received by the sender, it is a strong indicationthat at least one segment has been lost. The TCP sender willassume enough time has lapsed for all segments to beproperly re-ordered by the fact that the receiver had enoughtime to send three duplicate ACKs.

When three or more duplicate ACKs are received, thesender does not even wait for a retransmission timer toexpire before retransmitting the segment.

D. Fast recovery

Since the Fast Retransmit algorithm is used whenduplicate ACKs are being received, the TCP sender hasimplicit knowledge that there is data still flowing to thereceiver. The reason is because duplicate ACKs can only begenerated when a segment is received. This is a strongindication that serious network congestion may not exist andthat the lost segment was a rare event. So instead ofreducing the flow of data abruptly by going all the way intoSlow Start, the sender only enters Congestion Avoidance mode. Rather than start at window of one segment as inSlow Start mode, the sender resumes transmission with alarger window, incrementing as if in Congestion Avoidancemode. This allows for higher throughput under the conditionof only moderate congestion [5].

III. TCP WESTWOOD

TCP Westwood (TCPW) is a sender-side-only modification to TCP New Reno that is intended to better handle large bandwidth-delay product paths (large pipes), with potential packet loss due to transmission or other errors (leaky pipes), and with dynamic load (dynamic pipes).[6]

TCP Westwood relies on mining the ACK stream for information to help it better set the congestion control parameters: Slow Start Threshold (ssthresh), and Congestion Window (cwin).

In TCP Westwood, an "Eligible Rate" is estimated and used by the sender to update ssthresh and cwin upon loss indication, or during its "Agile Probing" phase, a proposed modification to the well-known Slow Start phase. In addition, a scheme called Persistent Non Congestion Detection (PNCD) has been devised to detect persistent lack of congestion and induce an Agile Probing phase to expeditiously utilize large dynamic bandwidth.[6]

The resultant performance gains in efficiency, without undue sacrifice of fairness, friendliness, and stability have been reported in numerous papers that can be found on The TCP WESTWOOD Home Page. Significant efficiency gains can be obtained for large leaky dynamic pipes, while maintaining fairness. Under a more appropriate criterion for friendliness, i.e. "opportunistic friendliness", TCP Westwood is shown to have good, and controllable, friendliness.

TCP Wetwood plus is an evolution of TCP Westwood, in fact it was soon discovered that the Westwood bandwidth estimation algorithm did not work well in the presence of reverse traffic due to ack compression. The TCP Westwood+ version is implemented in the Linux kernel.

The Fast Recovery algorithm from TCP New Reno has been modified. To help gain faster recovery bandwidth estimation (BWE) algorithm also has been added. This BWE function is what makes TCP Westwood standout. Influenced by TCP Vegas, BWE uses the RTT and the amount of data that has been sent during this interval to calculate an estimate of the currently successful transfer rate. The bandwidth estimate is then used when a loss is detected, setting cwnd and sssthresh at values near the estimation.

The main purpose behind this is to improve the throughput in wireless links, where loss is more often caused by link failure than by congestion. There is also the general benefit that starting CA at higher values will lower the recovery time on most networks, thus lowering the transfer times.

IV. TCP WESTWOOD END TO END BANDWIDTH MEASUREMENT

Sample of bandwidth used by that connection in TCP Westwood can be measured by
b2 = d2/ (t2-t1)
t1 is the time of previous ack that was received. An average of sample is calculated and used to calculate the estimation of available bandwidth.

A. Bandwidth estimation (BWE) algorithm

BWE =bk = α k bk−1 + (1 − α k) [(bk + bk−1) /2]
Where bk = sample bandwidth = dk / t k − t k−1
where dk = amount of bytes acknowledged by ACK k,
t k = arrival time of ACK k,
α k = [2τ − Δ(t k − t k−1)] / [2τ + Δ(t k − t k−1)]
where τ : is the cut- off frequency of this Tustin filter[7]

V. AN IMPROVED TCP WESTWOOD CONGESTION AVOIDANCE MECHANISM

The pseudo code of the TCP Westwood algorithm after three duplicate acknowledgements is: After 3 DUPACKS

If receiving 3 DUPACKS
Set ssthresh =(BWE*RTTmin) /seg_size;
and if cwnd>ssthresh then set cwnd = ssthresh ;
enter congestion avoidance

In the pseudo-code, seg_size indicates the length of TCP segments in bits. During the congestion avoidance phase the sender is trying for extra available bandwidth. If three duplicate ACKs are received, the network capacity might have been reached or that in case of wireless links, one or more segments have were dropped due to sporadic losses.[8]

A. Improved congestion avoidance algorithm

Slow Start is over */
/* cwnd>ssthresh */
Every Ack ;
Estimate BWE ;
Set BWE = BWlatest ;
BW = BWlatest/BWpast ;
If (1.5>BWratio>=1)
cwnd = cwnd + 2/cwnd ;
If (BWratio>= 1.5)
cwnd = cwnd + 4/cwnd ;
Else if (BWratio<1)
cwnd = cwnd + 0
Until (timeout or 3 DUPACKs)

In this method, the ratio of latest bandwidth and past bandwidth is calculated, if the ratio is greater than 1 and less than 1.5 than we increase congestion window by 2/cwnd, and if the ratio is greater than 1.5 then we increase congestion window bt 4/cwnd. The reason behind this is if the ratio is greater than 1, it means there is more bandwidth available to pass the packets. And if the ratio is less than 1, we will keep congestion window as it is.

VI. IMPLEMENTATION RESULTS

We have implemented this algorithm in wired and wireless topologies using ns2.35 and getting following results.

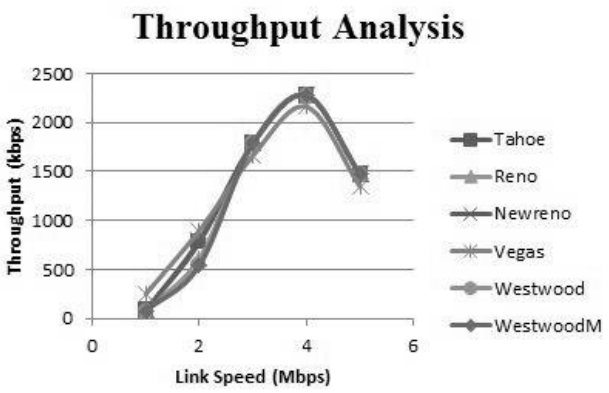

Fig.1: Throughput analysis in wired network

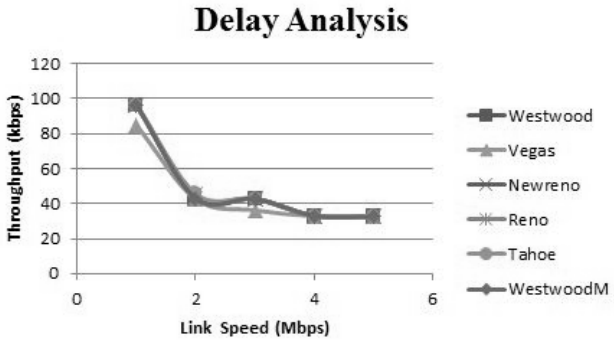

Fig.2: Delay analysis in wired network

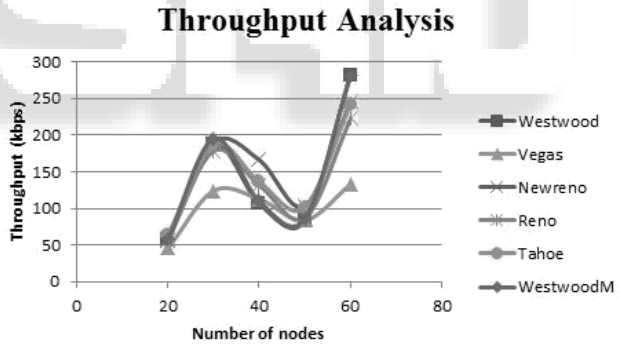

Fig.3: Throughput Analysis in wireless network

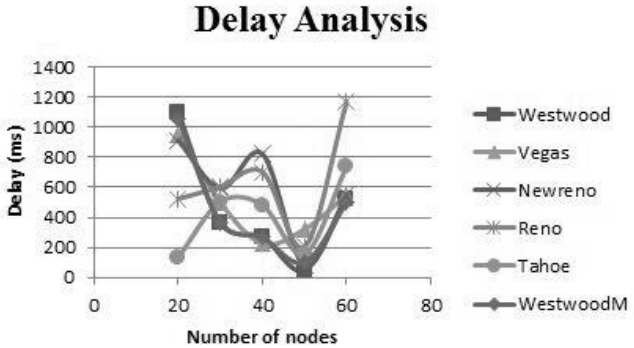

Fig.4: Delay Analysis in wireless network

VII. CONCLUSION

As we can see from the graphs that TCP Reno and TCP Newreno performs almost same, TCP Vegas performs better

for small link speed for throughput but when the link speed is increased than the performance of Vegas degrades for throughput. TCP Vegas not performs well than all other variants in terms of delay.TCP WestwoodM gives better result in terms of Throughput and delay in increasing number of nodes in wireless network. By applying the improved Congestion Avoidance method of TCP Westwood we can get better result in terms of throughput as well as delay.

ACKNOWLEDGMENT

My Sincere thanks to my guide Prof. Priyang Bhatt, for providing me an opportunity to do my research work. I express my thanks to my Institution namely G. H. Patel College of Engineering and Technology for providing me with a good environment and facilities like Internet, books, computers and all that as my source to complete this research work. My heart-felt thanks to my family, friends and colleagues who have helped me for the completion of this work.

REFERENCES

[1] C. E. Perkins, "Ad Hoc Networking", Addison - Wesley, Pearsonedu. Jan 2001.
[2] C. Callegari , S. Giordano, M. Pagano, T. Pepe, "Behavious analysis of TCP linux variants", ELSEVIER 2011
[3] K.Kathiravan, Dr. S. Thamarai Selvi, A.Selvam "Tcp Performance Analysis for mobile adhoc network using on demand routing protocols"
[4] JACOBSON, V. Congestion avoidance and control. In *ProceedingsOf SIGCOMM '88* (Stanford, CA, Aug. 1988), ACM.
[5] W. Stevens. TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms, January 1997, RFC 2001.
[6] Information on TCP Westwood en.wikipedia.org/wiki/TCP_Westwood
[7] Antonio Capone, Luigi Fratta, Fabio Martignom "An enhanced bandwidth estimation algorithms in the TCP congestion control scheme"
[8] ShimaaHagag,Ayman EI Sayed,"Enhanced TCP Westwood Congestion Avoidance Mechanism (TCP Westwood New)"
[9] MO. Richard and J.La, *Analysis and Comparison of TCPReno and Vegas*, INFOCOM'99. Berkeley, California:IEEE, 1999.
[10] R. Awdeh, Evaluating Fairness In Heterogeneous WirelessAd Hoc Networks, ICCCN'07. Sharjah, UAE: IEEE, 2007
[11] K. Fall and S. Floyd, Simulation-based comparisons of Tahoe, Reno, and SACK TCP, ACM SIGCOMM ComputerCommunication Review, Vol. 26, pp. 5-21, 1996.
[12] D. Kim and H. Bae., *Analysis of the Interaction betweenTCP Variants and Routing Protocols in MANETs*, ICPPW05,Oslo, Norway: IEEE, 2005.
[13] Shagufta Henna, "A Throughput Analysis of TCP Variants in MobileWireless Networks", IEEE 2009

