IJCSC International Journal of Computer Science & Communication Vol. 1, No. 2, July-December 2010, pp. 383-386

Simulation Base Analysis of TCP Reno and TCP Westwood Over IEEE 802.11 Wireless Ad hoc Networks

Dinesh C. Dobhal[1] & Rakesh Sharma[2]

[1]Dept. of Computer Application, Graphic Era University, Dehradun
[2]Dept. of Computer Science & Engineering, Graphic Era University, Dehradun
Email: [1]dineshdobhal@rediffmail.com

ABSTRACT

Mobile ad hoc networking has been a fast growing research area for the last few years. The need for a network when there is no fixed infrastructure is no more limited to military and emergency applications; ad hoc networks can include private and public applications as well. In ad hoc networks, wireless mobile computing devices can perform critical network topology functions that are normally the responsibility of the routers within the Internet infrastructure. Although there are many kinds of protocols available today that are supported by fixed network infrastructure, they need adaptation before they can be useful in ad hoc networks no longer connected to the Internet infrastructure. In this paper, we have studied the performance of TCP Westwood as compared with TCP Reno, to investigate the possibility of implementation of TCP Westwood in wireless environment and so in IEEE 802.11 wireless ad hoc networks.

Keywords: Mobile Ad hoc Network, TCP Reno, TCP Westwood

1. INTRODUCTION

TCP is a transport layer adaptive protocol which controls load offered to it by adjusting its window size according to the bandwidth of network [1] [2]. In case of no congestion, it additively increases its congestion window, whereas it throttles down its window when a sign of congestion is found. It is dominating transport layer protocol for best-effort traffic in the internet. It has been revised and extended several times over the past three decades, where most of the improvements were in congestion control mechanism.

TCP Westwood [3], another modification of TCP source protocol, was a modification to TCP to achieve fast recovery. As compared with TCP Reno, which halves the congestion window on receiving three duplicate acknowledgements, TCP Westwood attempts to make much more informed decision. As far as TCP Reno and TCP Westwood are concerned, both cannot distinguish between losses due to buffer overflow and random losses. If random loss is taken, TCP Reno overreacts and reduces the congestion window by half. Where as on the other hand, TCP Westwood resumes with the previous congestion window after a packet loss or retransmission timeout till the bottleneck link is not saturated.

In [4] several schemes have been proposed, to prevent the unnecessary reduction of congestion window by TCP Reno in case of random packet loss. All these schemes are based on support from intermediatory or proxy nodes. TCP Westwood does not require any cooperation from intermediatory nodes, thus preserving the "end-to-end design" principle [5].

To address TCP Reno friendliness another refinement of TCP Westwood is proposed by name TCPW RE (Rate Estimation) in [6]. To improve performance of TCP Westwood in internet, TCPW+ is described and studied in [7].

In this paper, we have studied the performance of TCP Westwood as compared with TCP Reno, to investigate the possibility of implementation of TCP Westwood in wireless environment and so in IEEE 802.11 static and mobile ad hoc networks. The remaining of the paper is organized in the following manner. *Section II* describes TCP Reno and TCP Westwood protocols. In *Section III*, simulation parameters and network topologies are described. Simulation results are analyzed in *Section IV*. In S*ection V*, conclusion and future work are discussed.

2. TCP RENO AND TCP WESTWOOD

[2] TCP Reno is the standard implementation of TCP. TCP Reno implementation includes slow-start, congestion-avoidance, fast retransmit and fast recovery algorithm. Both fast retransmit and fast recovery [8] are based on counting duplicate acknowledgements. [2] [9] fast retransmit and fast recovery are the algorithms designed to preserve self-clock during recovery from a lost segment. Duplicate acknowledgements are used to detect loss of a segment in fast retransmit algorithm. On

detection of three duplicate acknowledgements, TCP assume that a segment has been lost and retransmit it. Fast recovery algorithm estimates the outstanding data remained in the network by counting duplicate acknowledgements. It expands congestion window on each duplicate acknowledgement that is received, results in new data to be transmitted as congestion window becomes large enough.

During the recovery from one segment loss, the fast retransmit and fast recovery algorithm preserve TCP's self-clock and enable it to keep the network full, but in case of multiple segment losses, Reno is unlikely to recover fully, resulting in a timeout and subsequent slow-start [10]. In TCP Westwood, the TCP sender continuously monitors acknowledgements from TCP receiver and computes the current Rate Estimation (RE). Rate Estimation (RE) is based on the rate at which acknowledgements are received and on their payload i.e. RE is the rate that TCP is currently achieving. Thus, by definition, it must be the feasible rate. Duplicate acknowledgements and delayed acknowledgements are properly counted for computation of RE. On receiving three duplicate acknowledgement or a time out, the TCP sender adjust the cwnd (congestion window) and ssthresh (slow start threshold) based on RE. When an RE is obtained, TCP Westwood uses the following algorithm to adjust cwnd and ssthresh.

```
If (3 DUPACKS are received)
ssthresh = (RE * RTTmin)/seg_size;
If (cwnd >ssthresh) /*congestion avoid*/
cwnd=ssthresh;
Endif
Endif
If (coarse timeout expires)
cwnd = 1;
ssthresh = (RE * RTTmin)/seg_size;
If (ssthresh < 2)
ssthresh = 2;
Endif
Endif
```

3. SIMULATION PARAMETERS AND NETWORK TOPOLOGIES

This section describes the network environment and the parameters for the simulation. The results in this paper are carried out by using ns-2 network simulator [11]. The network topology used in the simulation is multi-hops topology, so that we may have multiple paths between sender A and the receiver I as depicted by the Figure 1.

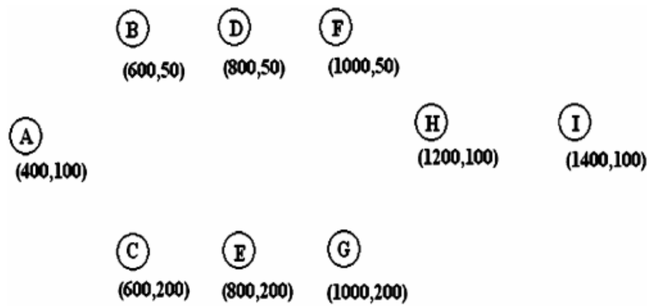

Fig. 1: Topology used in the Simulation

Simulation Parameters

Simulation Area	2000 x 2000 m
Simulation time	150 sim sec
Wireless nodes	9 (static)

Location of nodes

Node	Location/Coordinate
Node (A)	(400.0, 100.0, 0.0)
Node (B)	(600.0,50.0, 0.0)
Node (C)	(600.0, 200.0, 0.0)
Node (D)	(800.0,50.0, 0.0)
Node (E)	(800.0, 200.0, 0.0)
Node (F)	(1000.0, 50.0, 0.0)
Node (G)	(1000.0,200.0, 0.0)
Node (H)	(1200.0, 100.0, 0.0)
Node (I)	(1400.0, 100.0, 0.0)

Routing protocol	AODV
Interface queue type	DropTail/PriQueue
MAC	802.11
TCP source	node(A)
TCP destination	node(I)
TCP packet Size	28-1500 bytes
Application	FTP data
Antenna	Unity gain, omni-directional antennas to be centered in the node and 1.5 meters above it by following configurations

Antenna/OmniAntenna set X_ 0
Antenna/OmniAntenna set Y_ 0
Antenna/OmniAntenna set Z_ 1.5
Antenna/OmniAntenna set Gt_ 1.0
Antenna/OmniAntenna set Gr_ 1.0

Initialize the Shared Media interface with parameters to make it work like the 914MHz Lucent WaveLAN DSSS radio interface by following configurations

Phy/WirelessPhy set CPThresh_ 10.0
Phy/WirelessPhy set CSThresh_ 1.559e-11
Phy/WirelessPhy set RXThresh_ 3.652e-10
Phy/WirelessPhy set Rb_ 2*1e6
Phy/WirelessPhy set Pt_ 0.2818
Phy/WirelessPhy set freq_ 914e+6
Phy/WirelessPhy set L_ 1.0

4. SIMULATION RESULTS AND ANALYSIS

Simulation is carried out in ns2 for 150 sim sec in two different scenarios. In first scenario TCP Westwood and in second scenario TCP Reno is used with same topology (see figure 1), as transport layer protocol between sender and receiver.

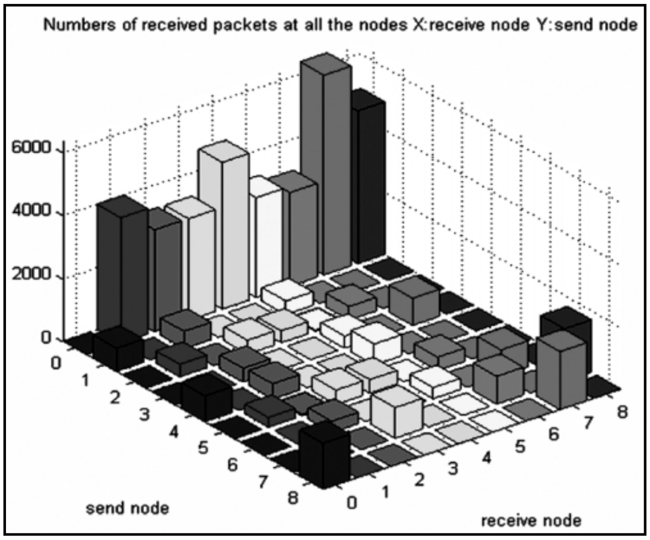

Fig. 2: Number of Received Packets at All Nodes using TCP Westwood

It is clear from figure 2, figure 3 and results collected that total number of packets received at node I are 6851 when using TCP Westwood. Total number of packets received at node I using TCP Reno are 5886 [figure 5 and figure 6]. End-to-End throughput of TCP Westwood is much more stable and high as compared with TCP Reno because of TCP Westwood introduces a “faster” recovery mechanism to avoid over-shrinking congestion window (cwin) after three duplicate ACKs [figure 4 and figure 7].

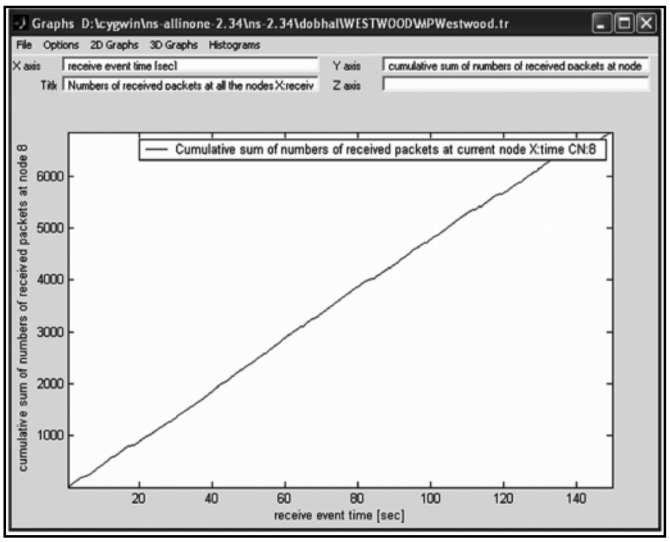

Fig. 3: End-to-End Cumulative Sum of Number of Packets Received at Node I Using TCP Westwood

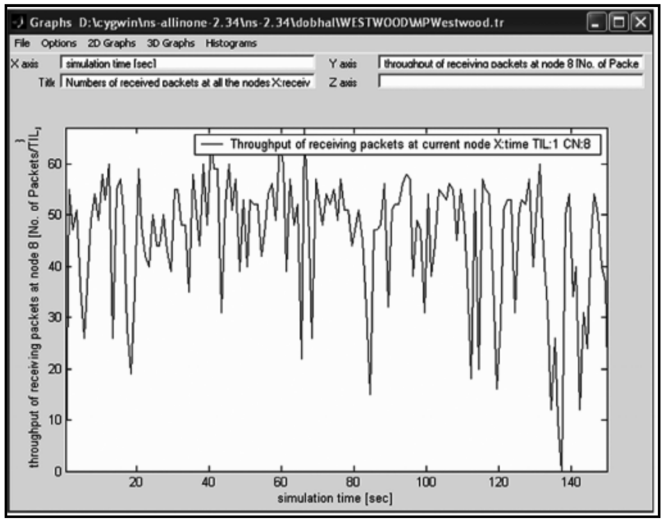

Fig. 4: Throughput of Received Packets at Sink [Node I] Using TCP Westwood

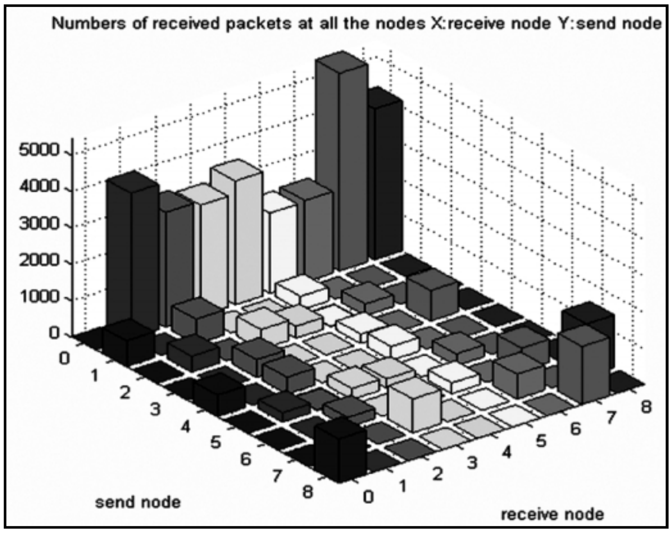

Fig. 5: Number of Received Packets at all Nodes using TCP Reno

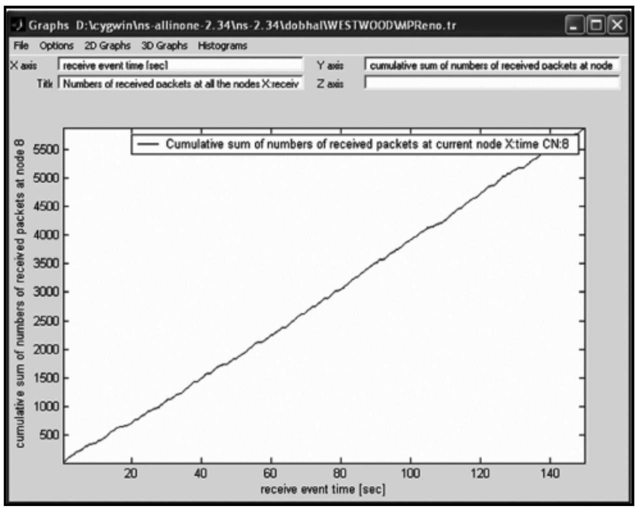

Fig. 6: End-to-End Cumulative Sum of Number of Packets Received at Node I Using TCP Reno

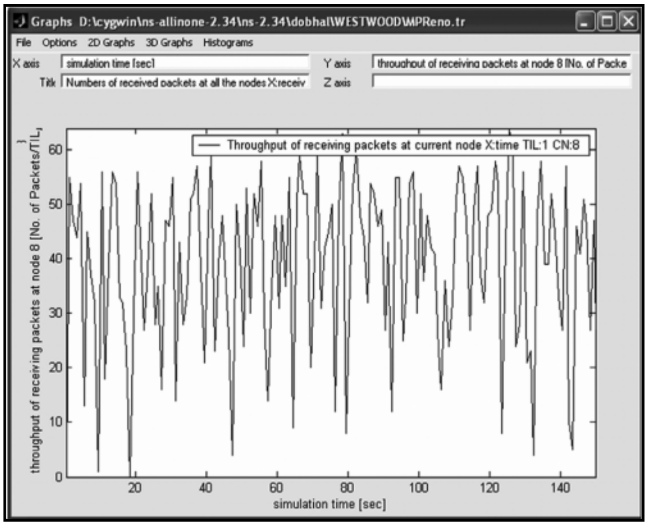

Fig. 7: Throughput of Received Packets at Sink [Node I] Using TCP Reno

5. CONCLUSION AND FUTURE WORK

In this paper we have compared TCP Westwood with TCP Reno through simulation, showing throughput gains in wireless scenarios. TCP Reno halves Congestion Window *(cwin)* after three duplicate ACKs where as TCP Westwood introduces a "faster" recovery mechanism to avoid over-shrinking *cwin* after three duplicate ACKs. It does so by taking into account the end-to-end estimation of the bandwidth available to TCP.

In the future, we will evaluate TCP Westwood in context of friendliness and more complex wireless topologies.

REFERENCES

[1] J. Postel, "Transmission Control Protocol", *Request for Comments 793, Internet Engineering Task Force*, September 1981.

[2] W. R. Stevens, "*TCP/IP Illustrated: Volume 1; The Protocols*", *Addison-Wesley Profesional Computing Series*, Addison-Wesley, Reading, Massachusetts, 1996.

[3] C. Casetti, M. Gerla, S. S. Lee, S. Mascolo, and M. Sanadidi, "TCP with Faster Recovery", In *IEEE Milcom*, 2000.

[4] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. H. Katz, "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links," *IEEE/ACM Transactions on Networking*, December 1997.

[5] M. Gerla, M. Y. Sanadidi, R. Wang, A. Zanella, C. Casetti, S. Mascolo, "TCP Westwood: Congestion Window Control Using Bandwidth Estimation", *In Proceedings of IEEE Globecom 2001*, **3**, pp1698-1702, San Antonio, Texas, USA, November 25-29, 2001.

[6] R. Wang, M Valla, M. Y. Sanadidi, M. Gerla, "Adaptive Bandwidth Share Estimation in TCPWestwood", *IEEE Globecom 2002*, Taipei, Taiwan, November 18-20, 2002.

[7] R. Ferorelli, L. A. Grieco, S. Mascolo, G. Piscitelli, P. Camarda, "Live Internet Measurements using Westwood+ TCP Congestion Control", *IEEE Globecom 2002*, Taipei, Taiwan, November 18-20, 2002.

[8] W. Richard Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms", March 1996, Currently an Internet Draft: draft-stevens-tcpca-spec-01.txt.

[9] Van Jacobson, Fast Retransmit, Message to the End2end-interest Mailing List, April 1990.

[10] Sally Floyd, "TCP and Successive Fast Retransmits", Feb 1995.