

Highbrow TCP as an Enhanced TCP New Reno for MANETs – A Simulation Approach

Hardik K. Molia
Rajkot, Gujarat, India

Rashmi Agrawal
Rajkot, Gujarat, India

ABSTRACT

The TCP – Transmission Control Protocol provides connection oriented and reliable process to process communication. TCP also performs flow control, error control and congestion control. TCP's default interpretation of any loss as a side effect of network congestion works well with the wired networks. But when Standard TCP is used with the wireless networks, especially with the MANETs - Mobile Adhoc Networks, it becomes a reason of performance degradation. MANETs may suffer from channel losses, route failure losses and congestion losses. Unfortunately it is not possible to differentiate these losses without predictions. Highbrow TCP is a probabilistic approach to predicate losses before TCP's conventional mechanisms are triggered for congestion control. Highbrow TCP doesn't differentiate losses but it can be used to avoid pauses in communications by redundant retransmissions. This paper is simulation based comparison of TCP NewReno and TCP NewReno with Highbrow TCP scheme. These two variants are implemented in NS-2.35 and have been tested over 3 different scenarios of MANETs. Performance is measured in terms of average throughput, instantaneous throughput and goodput with various transmission rates.

General Terms

Wireless Adhoc Networks, Mobile Adhoc Networks, Congestion Control, Loss Differentiation Algorithms

Keywords

TCP, MANET, TCP NewReno, Highbrow TCP, Fastest Retransmission, Throughput, Goodput.

1. INTRODUCTION

Standard TCP controls the transmission rate of a process to process communication by maintaining a byte oriented sender window whose size is the minimum of the rate at which receiver can receive as well as the network can handle. Receiver informs its desired receiving rate as a part of TCP header in the field window size. TCP Sender approximates the network congestion by using various congestion control schemes like AIMD – Additive Increase, Multiplicative Decrease scheme. Based on the estimated congestion, Sender TCP decides the rate at which further data should be transmitted. Thus the efficiency in estimation of network congestion plays a major role in overall performance of the communication [1].

Standard TCP was initially used with the wired networks where the probability of congestion loss is much higher than of the probability of random channel loss and route failure loss. The TCP theory suited well and improved performance significantly too. But when standard TCP is used with the wireless networks, especially with the MANETs, TCP's default consideration of any loss as a congestion loss degrades the overall performance. As MANETs often suffer from channel loss and route failure loss, it is required to modify the

TCP to make it capable of distinguishing various losses and act accordingly. Ideally TCP should slow down the rate on congestion loss, retransmit the lost packets on channel loss and wait till the new route is established on route failure loss. But unfortunately, TCP has no capability to distinguish losses. An enhanced scheme needs to predicate the possibility of losses and approximate the amount of losses [1]. Many variations have been proposed which are based on predictions. Highbrow TCP is such a novel scheme which we proposed as a rough sketch earlier [6]. This paper shows how it can be adopted with TCP NewReno for the performance improvement in MANETs.

2. TCP NEWRENO

2.1 Sender Sliding Window

TCP keeps two sliding windows, Cwnd-congestion window and Rwnd-receiver's advertised window. Sender estimates Cwnd as per the network congestion. Receiver informs the upper bound of its receiving rate through TCP header's Window Size field in the form of Rwnd. Sender selects the minimum of Cwnd and Rwnd as the transmission rate [2][3][4].

$$W_{nd} = \min(C_{wnd}, R_{wnd})$$

Sender window supports open, close and shrink operations while receiver window supports open and close operation based on basis of sliding window concept.

2.2 Slow Start – Exponential Increase

TCP starts communication with initially small Cwnd, 1 or 2 MSS – Sender Maximum Segment Size. Slow start increases the size of Cwnd by 1 with every successfully received ACK - Acknowledgement. So after every RTT-Round Trip Time, Cwnd gets doubled. TCP sets ssthresh – Slow Start Threshold value. TCP continues in Slow Start until $C_{wnd} \geq ssthresh$ [2][3][4].

2.3 Congestion Avoidance

TCP enters into congestion avoidance once Cwnd becomes greater than or equal to ssthresh. TCP still continues to increase the sending rate by increasing Cwnd by 1 with every RTT, so it is called additive increase TCP continues in this phase until RTO-Retransmission Timer times out or three duplicate acknowledgements are received. On retransmission times out, TCP sets ssthresh to half of the Cwnd, sets Cwnd to 1 and starts slow start phase again [2][3][4].

2.4 Fast Retransmission and Fast Recovery

It is possible to detect congestion even before RTO-Retransmission Timer times out. Whenever receiver receives an out of order packet, it sends a duplicate ACK immediately. Sender counts the number of such duplicate ACKs. On receiving 3 same duplicate ACKs, TCP considers a packet loss and switches to Fast Retransmission Phase. Fast retransmission retransmits the oldest unacknowledged packet and enters into the Fast Recovery phase. TCP stays in the fast

recovery phase until all the outstanding packets are acknowledged successfully and then enters into the congestion avoidance phase [2][3][4].

3. FASTEST RETRANSMISSION PHASE

A novel Fastest Retransmission Phase is introduced in [5]. If TCP sender is waiting for an ACK for time up to T1 and has not received anything yet, TCP sender retransmits the oldest unacknowledged packet without waiting for three duplicate ACKs or RTO timeout conditions. This scheme is useful when the amount of possible outstanding packets is small and so sender may not receive three duplicate ACKs due to loss of few of them [5]. T1 is the Fastest Retransmission Timer which is set as,

$$T1 = \frac{1}{2} RTT \quad \text{Where } (\alpha * RTO) < \frac{1}{2} RTT$$

$$T1 = \alpha * RTO \quad \text{Where } \frac{1}{2} RTT \leq (\alpha * RTO) < \frac{3}{2} RTT$$

$$T1 = \frac{3}{2} RTT \quad \text{Where } (\alpha * RTO) > \frac{3}{2} RTT$$

T1 is a function of $\alpha * RTO$ where the proposed value of α is 1/5 [5].

4. Highbrow TCP

In existing system [5], on expiration of Fastest Retransmission Timer, T1, Fastest Retransmission Phase always retransmits the oldest unacknowledged packet. We proposed an enhanced fastest retransmission phase in [6] which is named as Highbrow TCP. A person is called to be highbrow if he has intellectuality as well as intelligence. Being Intellectual, highbrow TCP analyses past few RTT to measure network instability. Being intelligent, highbrow TCP takes decision from the measured instability about how many oldest unacknowledged packets should be retransmitted [6].

4.1 Sender Side Algorithm

On time out of fastest retransmission timer T1, FRP - Fastest Retransmission Phase is activated. On completion of retransmission, the timer T1 is reset and TCP continuous with the traditional mechanism. FRP can be described algorithmically as follow [6].

1. Find Mean value of last 10 RTTs
2. Find Variance.
3. Find Standard Deviation.
4. Find Relative Standard Deviation - RSD
5. Number of outstanding unacknowledged packets are M
6. $Temp = 100 - RSD$
7. if ($Temp > 0$)
Retransmit first $Temp\%$ packets out of M outstanding unacknowledged Packets. Do round off when needed.
8. Reset Fastest Retransmission Timer T1.
9. Exit

Flow chart of Highbrow TCP's Fastest Retransmission is shown in figure 1 [6].

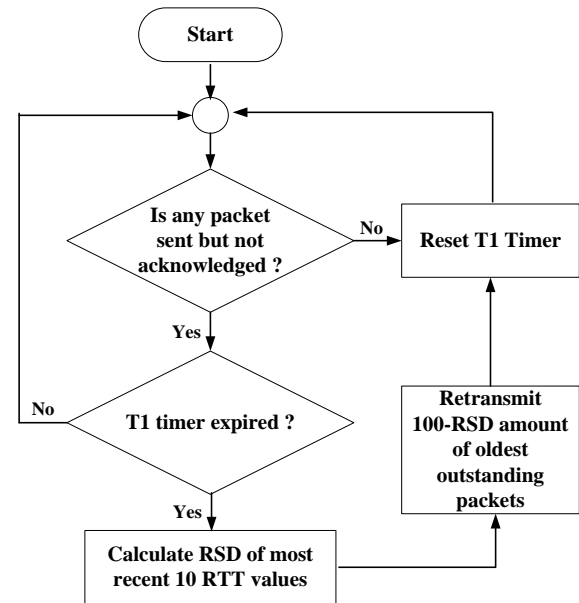


Figure 1. Highbrow TCP's Fastest Retransmission Scheme

5. SIMULATION

Simulation of Highbrow TCP is performed with NS – 2.35. It has been tested with Scenario based MANETs and compared with existing TCP NewReno.

5.1 Scenario based MANETs

Three scenarios are designed to implement MANET. Scenarios differ by number of nodes, mobility speed and mobility redefine interval. Simulation time is taken as of 50 seconds for all scenarios. Mobility redefine interval is 5 seconds for all scenarios [7].

Table - 1 MANET Scenarios

Scenario	Nodes	Mobility Speed (Seconds)
MANET-M1	5	20
MANET-M2	7	25
MANET-M3	10	20

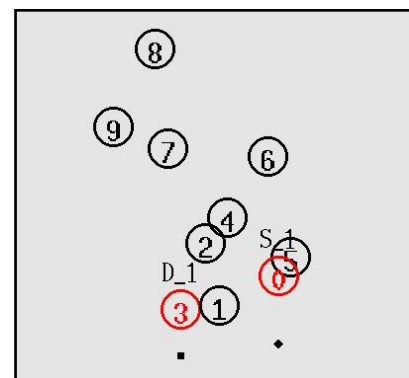


Figure 2. MANET Implementation in NS 2.35

5.2 Average Throughput

Average Throughput for TCP NewReno and Highbrow TCP are measured in Kbps – Kilo Bits per Second for all the three MANET scenarios. Results are shown in Figure 3.

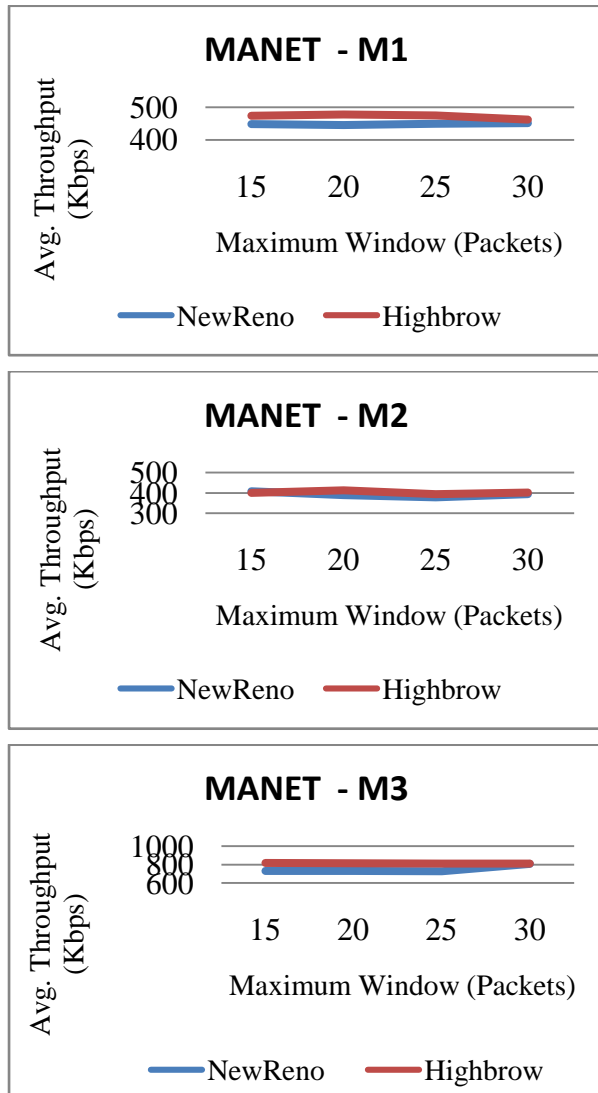


Figure 3. Average Throughput

5.3 Instantaneous Throughput

Instantaneous Throughput for TCP NewReno and Highbrow TCP are measured with interval of 5 seconds and with window of 10 packets. Results are shown in Figure 4.

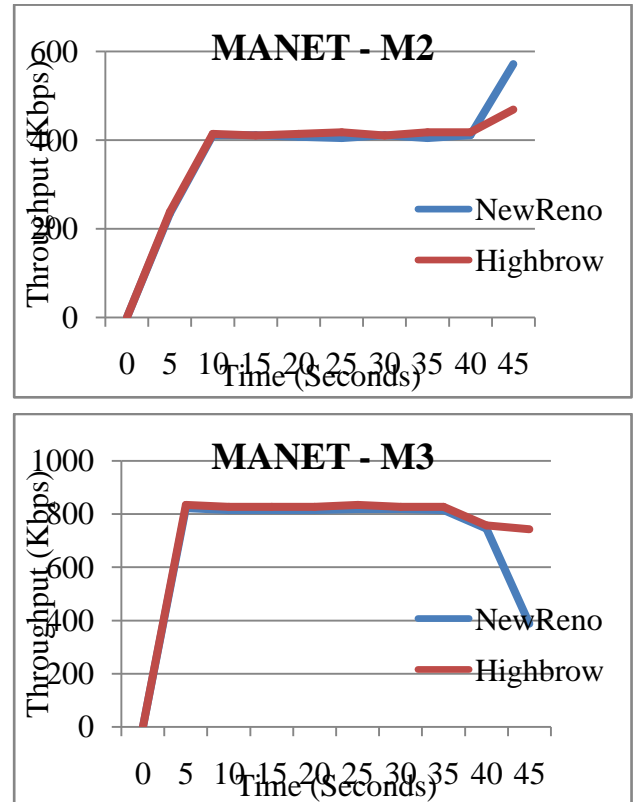
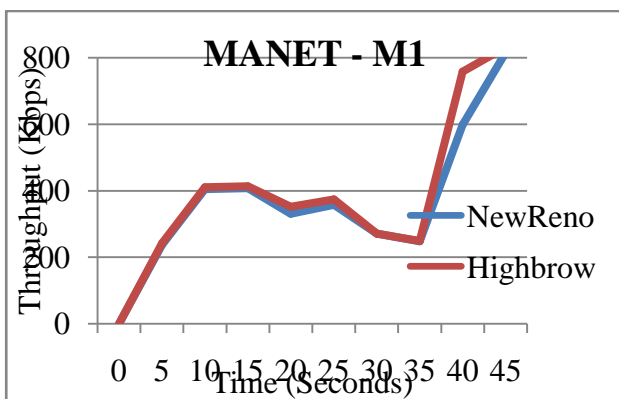
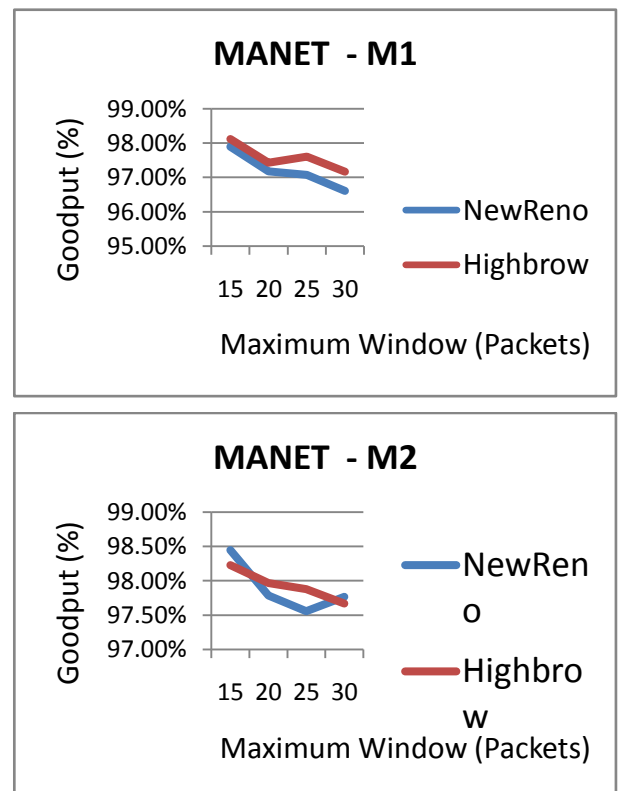


Figure 4. Instantaneous Throughput

5.4 Goodput

Goodput for TCP NewReno and Highbrow TCP are measured as the ratio of number of received packets and number of generated packets as shown in Figure 5.



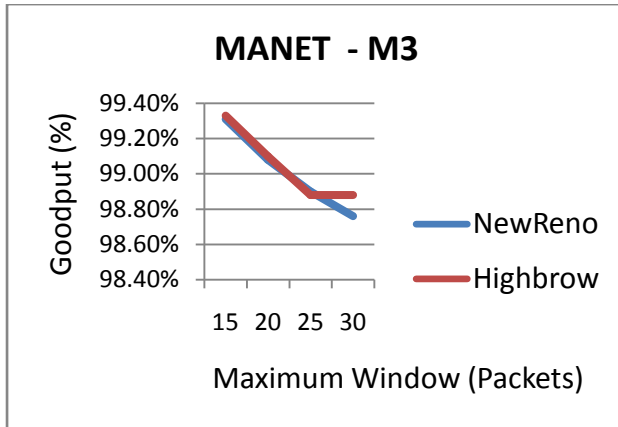


Figure 6. Goodput

Table - 1 Simulation Results

TCP NewReno		Highbrow TCP		
Max. Window	Avg. Throughput	Goodput	Avg. Throughput	Goodput
MANET – M1				
15	448	97.90%	474	98.12%
20	445	97.18%	478	97.43%
25	449	97.08%	475	97.61%
30	451	96.61%	462	97.17%
MANET – M2				
15	407	98.45%	400	98.23%
20	388	97.79%	412	97.97%
25	377	97.56%	393	97.88%
30	392	97.77%	401	97.67%
MANET – M3				
15	728	99.31%	816	99.33%
20	727	99.08%	810	99.10%
25	725	98.90%	808	98.88%
30	804	98.76%	809	98.88%

6. CONCLUSION

Simulation of Highbrow TCP is performed with NS – 2.35. It has been tested with Scenario based MANETs and compared with existing TCP NewReno. Highbrow TCP has an additional feature of Fastest Retransmission Phase over the traditional fast retransmission and RTO time out based retransmissions. In the simulation, it has shown significant amount of performance improvement over existing TCP NewReno.

IEEE 802.11 WLAN based MANETs have maximum of 2264 Bytes as a single packet size. The transmission rate doesn't grow beyond 10 Kbytes. Maximum number of outstanding packets at any moment could be 4 only. It is natural that out of these 4 packets, if 1 or more packets are lost due to any reason, sender will not able to receive 3 duplicate ACKs. Subsequently RTO will time out and TCP starts slow start phase. This unnecessary reduction of transmission rate and pauses in transmission can be handled by adapting Highbrow TCP where TCP tries to predicate losses in between of fast retransmission and RTO times out. If network is stable, sender gets ACKs in time and there. This scheme can be implemented with any of the TCP reactive variants.

7. REFERENCES

- [1] Subir Kumar Sarkar, T G Basavaraju, C Puttamadappa , "Ad Hoc Mobile Wireless Networks - Principles, Protocols, and Applications" - Auerbach Publications Taylor & Francis Group
- [2] Behrouz Forouzan, "TCP/IP Protocol Suite", 4/e, McGraw-Hill
- [3] W. Richard Stevens, "TCP/IP Illustrated, Vol. 1: The Protocols" (Addison-Wesley Professional Computing Series)
- [4] Ghassan A. Abed *, Mahamod Ismail, Kasmiran Jumari, "Exploration and evaluation of traditional TCP congestion control techniques", Journal of King Saud University – Computer and Information Sciences (2012) 24, 145–155
- [5] Sungkwan Jung, Jeongman Lee, Gangminh Lee, Sung-Yeop Pyun, Dong-Ho Cho, "Novel Fastest Retransmission and Rate Control Schemes for Improving TCP Performance in Wireless Ad Hoc Networks" Springer Science+Business Media New York 2013, Wireless Pers Commun (2014) 75:557–567
- [6] Molia, Hardik K. ; Agrawal, Rashmi , "Highbrow TCP - A Statistical Sketch for Preventing Pauses in Wireless Transmissions" , Electronics, Communication and Computing Technologies (ICECCT), 2015 International Conference on
- [7] Teerawat Issariyakul, Ekram Hossain, "Introduction to Network Simulator NS2" , Springer; 2nd ed. 2012 edition
- [8] Molia, Hardik K. ; Agrawal, Rashmi , "A conceptual exploration of TCP variants" Emerging Technology Trends in Electronics, Communication and Networking (ET2ECN), 2014 2nd International Conference on