ISSN:0975-9646

Poonam Tomar et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 2 (5) , 2011, 2467-2471

A Comprehensive Analysis and Comparison of TCP Tahoe, TCP Reno and TCP Lite

Poonam Tomar, Prashant Panse

Department of Information Technology, Mahakal Institute of Technology
Ujjain (M.P.), India

***Abstract*— There are various TCP variants and each one belongs to a different criteria. In this paper we discuss about the congestion problem in Adhoc networks and compare the performance of three TCP variants that all work on different techniques. This paper compares TCP variants specifically TCP Tahoe, Reno and Lite based on different parameters such as number of nodes received with error, packet loss, byte received, and throughput and pause time. A table is then drawn which shows the comparison results.**

***Keywords*— congestion; Adhoc Networks; TCP variants;**

I. Introduction

Mobile Adhoc Networks (MANET) is some complex systems which work as collection of mobile users to provide communication over wireless links. A MANET does not have a fixed infrastructure. Nodes are mobile so topology keeps on changing. They are applicable in such situations where no infrastructure is available. Most of the MANET applications make use of a reliable end to end transport protocol such as TCP [1],[2],[3] include to set up an end to end connection for end to end delivery of data packets, flow control and congestion control. TCP has proved to perform reliably in traditional wired and stationary networks where the main reason for losses in network congestion but it does not perform as so when applied to wireless networks. It is because of the misinterpretation of the losses that are not caused by network congestion. Unfortunately it invokes a congestion control algorithm which reduces the bandwidth utilization and become the reason for performance degradation by providing poor throughput and higher delays.

Some of the other reasons behind throughput degradation [4] are:

1. Misinterpretation of packet loss
2. Frequent path breaks
3. Misinterpretation of congestion window
4. Multipath Routing
5. Network partitioning and emerging
6. The use of Sliding Window based transmission.
7. Effect of the path length.
8. Asymmetric link behaviour
9. Unidirectional

For all the above based reasons congestion control is thought to be an important issue for Manet's. Several protocols have been suggested time to time for providing a solution to the congestion problem. These are called TCP Variants. Traditional TCP is known as TCP Tahoe, after that came different versions of TCP like Reno, New Reno, SACK, FACK, Vegas, and Lite. These are all called variants of TCP because each type possesses some special criteria. TCP Reno is advance version of TCP Tahoe [5] with fast recovery added to it. TCP New Reno applies the newest retransmission mechanism to TCP Reno. TCP SACK provides the facility to specify several additional data packets that have been received out of order within one duplicate acknowledgment(DUPACK) instead of only the last in order packet received [7].TCP Vegas proposes some unique retransmission and congestion control strategies. When Forward ACK [8] is applied to Reno it becomes TCP FACK.

In this paper we are making out a comparison of three TCP Variants TCP Tahoe, TCP Reno and TCP Lite. These three Variants are chosen for comparison because all the three work on different strategies.

Rest of this paper is structured as follows. In section I we will give a short introduction to the congestion problem in general. In section II we will describe TCP Variants: TCP Tahoe, Reno and Lite. In section III we will compare these Variants on qualnet simulator based on different parameters and simulation results will be shown. Finally some general observations and concluding remarks follows in section IV and an outlook on possible future research directions is given in section V.

II. Congestion Problem

When there are several resources in a network that are shared by multiple competing senders it becomes difficult to manage the data rate used by each sender so that network need not be overloaded. The network congestion can cause severe degradation of throughput. If no proper approach is followed for controlling the congestion than it can even collapse the network.

A. TCP Congestion Control

In Internet congestion control is the responsibility of Transport Layer more precisely of the Transmission Control Protocol (TCP). TCP combines congestion control and reliability mechanisms. To detect network congestion TCP simply observes occurring packet losses. Since on the Internet missing packets are almost always caused by congestion, a missing packet is interpreted as a sign for network congestion. TCP uses cumulative acknowledgments: a TCP receiver always acknowledges the end of the so-far correctly and completely received data when a new segment arrives. If segments are received out-of-order, i. e., some data is missing between the already known and the newly arriving data, the last acknowledgment is sent again (duplicate ACK). In TCP a window-based additive increase, multiplicative decrease mechanism is employed. The window size is increased by one segment in every round-trip time when no packet losses occur. In case of the reception of a duplicate acknowledgment a TCP sender will first assume that some packet reordering has occurred in the network. But upon reception of the fourth copy of an acknowledgment (Triple Duplicate ACK, TDACK) a congestion loss is assumed. In this case the missing segment is repeated and the window size is cut in half (multiplicative decrease).

Additionally, TCP uses a timeout that depends on the measured round-trip time of the connection. If this retransmission timeout (RTO) elapses without an acknowledgment TCP concludes severe congestion. Then the window size is reduced to one and the unacknowledged segment is sent again. The timeout until the next retransmission attempt if still no acknowledgment arrives is doubled. Thus this timeout grows exponentially. During the first phase of a connection and after a timeout a mechanism named slow start is employed. It allows for a faster convergence to the correct window size. While slow start is active, the window size is not increased by one segment size for every round-trip time, but instead for every received acknowledgment. This means that during this phase the window size grows exponentially.

B. Congestion In Mobile Adhoc Networks

TCP Congestion control works well for Internet but since Manet's exhibit some unique properties congestion control in Manet's is problematic for standard TCP. Important amongst the specific properties of Manets are the node mobility and a shared, wireless multi-hop channel. Route changes due to node mobility as well as the inherently unreliable medium results in unsteady packet delivery delays and packet losses. These delays and losses must not be misinterpreted as congestion losses. due to the comparatively low bandwidth of mobile ad-hoc networks, one single sender causes a collapse of the network due to congestion. The extreme effect of a single traffic flow on the network condition can cause severe unfairness between flows. Thus wireless multihop networks are much more prone to overload-related problems than traditional wire line networks like the Internet. Therefore an appropriate congestion control is required for network stability and acceptable performance.

III. DESCRIPTION OF TCP VARIENTS

There are many versions of TCP which have been modified time to time as per need. In the earliest TCP, there were limited facilities to minimize the network congestion. Implementation used cumulative positive acknowledgements and the expiry of a retransmit timer to provide reliability based on a simple go-back-n model. Several succeeding versions of TCP based on congestion control and avoidance mechanism have been developed since then. In this section, we will discuss the performance of various TCP versions like Tahoe, Reno and Lite.

A. TCP Tahoe

TCP Tahoe is one of the congestion control algorithms described adds some new and enhance the earlier TCP implementations including Slow Start, Congestion Avoidance and Fast Retransmit. It works on the packet conservation policy. Which is 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? This policy is implemented by using acknowledgments to clock outgoing packets. It also maintains a congestion window (CWD) to reflect the network capacity.

The shortcoming in TCP Tahoe is that packet loss is detected after the whole timeout interval. When the packet loss is detected, TCP Tahoe's performance becomes slow. Due to this reason transmission flow decreases.

B. TCP Reno

In TCP Reno[9] after the first retransmit the communication path pipe does not gets empty as in TCP Tahoe. And thus it avoids slow start to fill it again after a packet loss. When a single packet is lost from a window of data, Reno maintains it by Fast Recovery mechanism but when multiple packets are lost Reno has same performance as Tahoe.

In Reno when three DUPACK are received it is assumed that segment was lost and that segment is transmitted without waiting for timeout. Another thing that is important in Reno is that it does not reduce the congestion window to 1 because it empties the pipe rather it applies Fast Retransmit.

The shortcoming with TCP Reno is it does not perform well in case of multiple packet losses since they are difficult to detect. The use of Coarse grained timer for RTT estimation results in poor performance.

C. TCP Lite

It is a service that provides a transport channel that intercepts TCP to reduce the overhead in session management in which no application data is transmitted or received. It reduces or eliminates pure TCP protocol data units which are used in the setup, teardown and acknowledgment of a channel while maintaining order, integrity, reliability and security of the original TCP transport.

When it comes to congestion control, TCP Lite doesn't have many advantages over Reno. It is similar to TCP Reno. It detects and re-transmits more than one lost packet before timeout occurs. It can suffer from performance drawbacks since packets are dropped in large amount. It has better congestion avoidance and bandwidth utilization over Tahoe and Reno because TCP Lite provides big window and protection against wrapped sequence numbers option, but it does not reduce the congestion window like that of Reno for congestion avoidance. It suggests better way for fast retransmission when packet losses in network.

IV. COMPARISON OF TCP VARIENTS

A. *Simulation Environment*

All the simulation work is carried out using TCP variants (Reno, Lite, Tahoe) with DSR routing protocol .Network traffic is provided by using File Transfer Protocol (FTP) application. File Transfer Protocol (FTP) represents the File Transfer Protocol server and client. Wireless network which we have used have following values for different parameter:

TABLE 1: SIMULATION ENVIRONMENT

Mobility model Random Way Point
Minimum speed 0 mps
Maximum 30 mps
Pause time 5s, 10s, 15s, 20s, 25s, 30s.
Simulation Time 200s
Terrain
Coordination 1500 * 1500 m
Connection
FTP (File transfer protocol): 41 (client) to 1 (server)
Item size 512(byte)
Radio/physical layer parameters*:*
Radio type: 802.11b Radio
Data rate: 2Mbps
Packet reception model: Bit error rate (bpsk.ber)
MAC Protocol*:* 802.11
Routing Protocol*:* DSR (Dynamic Source Routing)
Transport Protocol*:* TCP Tahoe, TCP Reno, TCP Lite,
Node*:* 50
Node Placement*:* Random
Seed: 1

B. *Simulation Methodology*

Performance metrics used for this works are as follows:

Throughput: It is the measure of the number of packets successfully transmitted to their final destination per unit time. It is the ratio between the numbers of sent packets vs. received packets

Signal Received with error: It is the measure of signal received, but they have error. The error may be occurring due to noise or due to heavy traffic.

Bytes received: They are the measure of total packet received by server. The packets may be drop due to heavy traffic.

Packet loss: It is the measure of total discarded packet due to corruption or due to packet drop. It is calculated by subtracting total received packets by server in total sent packet by client.

TABLE 2: SIMULATION METHODOLOGY

Sr. No.	Parameters	Value
1.	Simulation Time	Constant
2.	No. of Node	Will Change
3.	Area	Constant
4.	Pause Time	Constant
5.	TCP Protocol	Will Change
6.	Routing Protocol	Constant
7.	Node Speed	Constant

C. *Results*

Our analysis of the result guides us to conclude that:

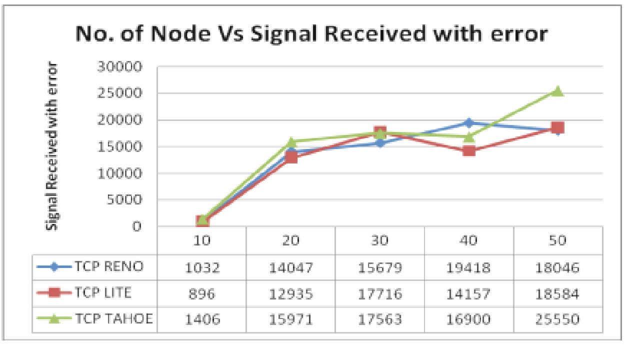
-- When increasing the no of nodes then the ratio of signals received with error keeps on increasing in case of all the three but then decreases in case of TCP Reno.

--The no of packet losses vary with the increase in no of nodes. TCP Lite shows a little increase in packet loss at start remains sometime with less loss and again starts to increase, TCP Reno shows a quick rise and fall in this case and TCP Tahoe performs average in this case.

-- TCP Lite shows a constant behavior in case of received bytes when no of nodes is varied but TCP Reno shows quick ups and downs. While TCP Tahoe also becomes constant after some time.

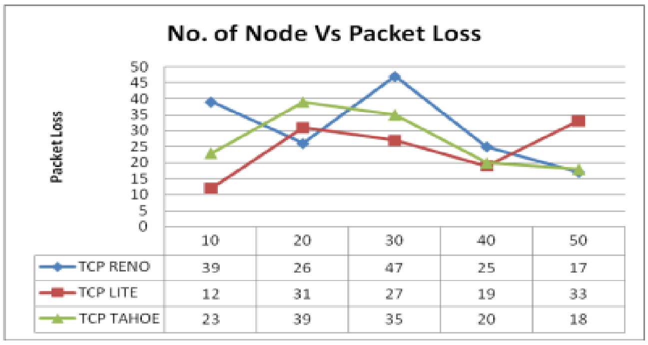
--TCP Tahoe provides better results in case of throughput than TCP Lite. Throughput is also average in case of TCP Reno.

--Average packet loss is less in case of TCP Tahoe but as the no of nodes increase packet loss increase in case of TCP Lite and Tahoe, but sudden decrease is seen in case of TCP Reno.



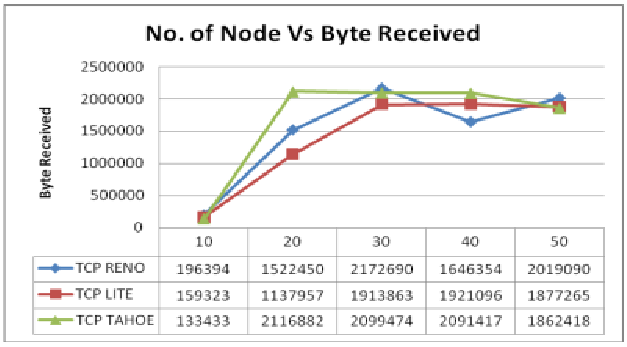
	10	20	30	40	50
TCP RENO	1032	14047	15679	19418	18046
TCP LITE	896	12935	17716	14157	18584
TCP TAHOE	1406	15971	17563	16900	25550

Fig 7.1 No of Nodes vs. Signal received with error



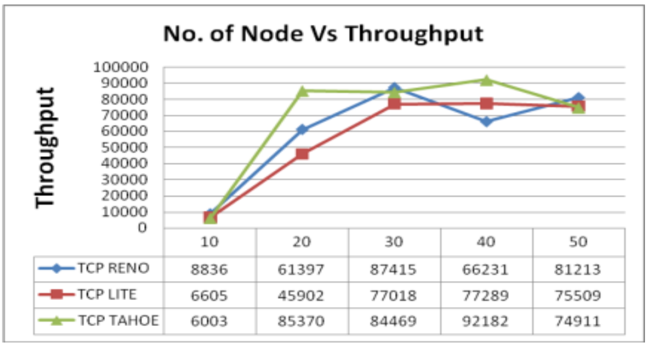
	10	20	30	40	50
TCP RENO	39	26	47	25	17
TCP LITE	12	31	27	19	33
TCP TAHOE	23	39	35	20	18

Fig 7.2 No of Node vs. Packet Loss



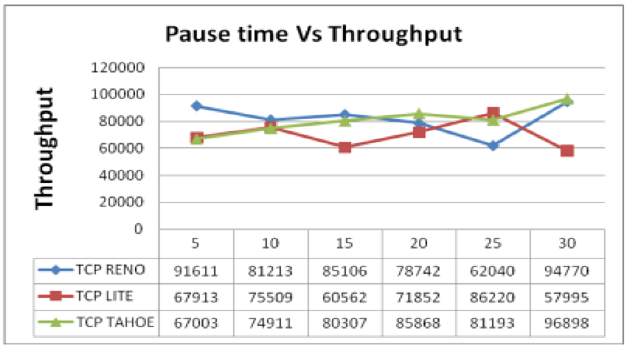
	10	20	30	40	50
TCP RENO	196394	1522450	2172690	1646354	2019090
TCP LITE	159323	1137957	1913863	1921096	1877265
TCP TAHOE	133433	2116882	2099474	2091417	1862418

Fig 7.3. No of Nodes vs. Byte Received



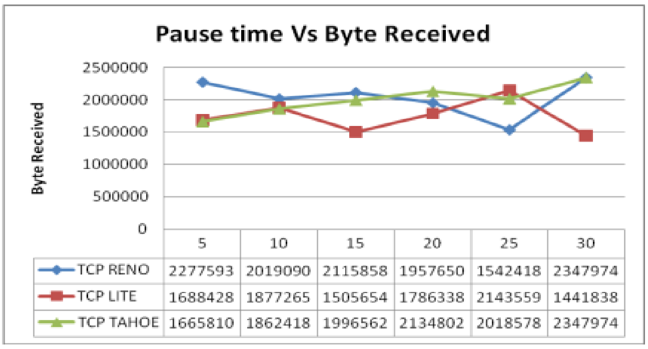
	10	20	30	40	50
TCP RENO	8836	61397	87415	66231	81213
TCP LITE	6605	45902	77018	77289	75509
TCP TAHOE	6003	85370	84469	92182	74911

Fig 7.4 No of Node vs. Throughput



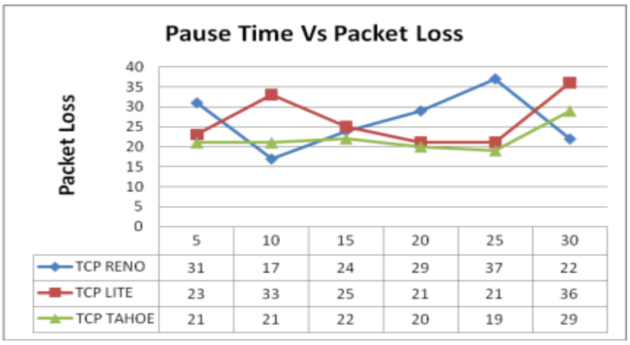
	5	10	15	20	25	30
TCP RENO	91611	81213	85106	78742	62040	94770
TCP LITE	67913	75509	60562	71852	86220	57995
TCP TAHOE	67003	74911	80307	85868	81193	96898

Fig 7.5 Pause time vs. Throughput



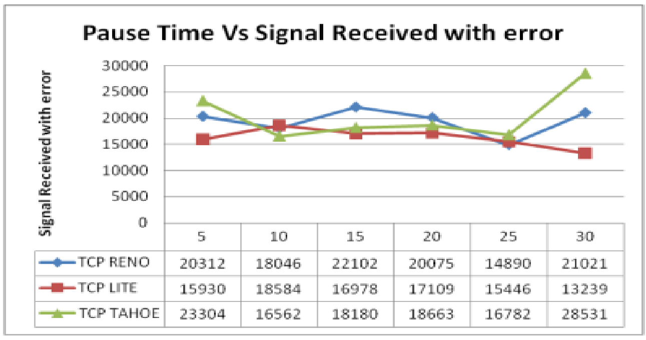
	5	10	15	20	25	30
TCP RENO	2277593	2019090	2115858	1957650	1542418	2347974
TCP LITE	1688428	1877265	1505654	1786338	2143559	1441838
TCP TAHOE	1665810	1862418	1996562	2134802	2018578	2347974

Fig 7.6 Pause time vs. Byte Received



	5	10	15	20	25	30
TCP RENO	31	17	24	29	37	22
TCP LITE	23	33	25	21	21	36
TCP TAHOE	21	21	22	20	19	29

Fig 7.7 Pause time vs. Packet Loss



	5	10	15	20	25	30
TCP RENO	20312	18046	22102	20075	14890	21021
TCP LITE	15930	18584	16978	17109	15446	13239
TCP TAHOE	23304	16562	18180	18663	16782	28531

Fig 7.8 Pause time vs. Signal Received with error

V. CONCLUSION

Congestion Control is a significant issue in Mobile Ad hoc Networks. The objectives listed in the problem statement have been carried out properly. In the presented work, all the simulation work was carried out using TCP variants (Reno, Lite, Tahoe) with DSR routing protocol .Network traffic is provided by using File Transfer Protocol (FTP) application. Everything is studied against various parameters such as, throughput, signal received with error, bytes received and packet loss while increasing the number of nodes. We sincerely hope that our work will contribute in providing further research directions in the area of routing.

Some of protocols show better response and some of them show poor responsiveness to changing network conditions and network utilization. Although there are various protocols and algorithms that have been used, there is no single algorithm that can overcome the congested and unreliable nature of network. In short, any protocol will be effective based on the parameters that are to be taken into consideration.

To conclude this area is not completely explored to it maximum and still lot more research can be done towards establishing a basis for the development of new protocols.

VI. FUTURE WORK

In the presented work, we have made a simulation study, it would be interesting to note and analyse the behaviour of a MANET on a real-life test-bed. We have seen that TCP Reno performs better in case of throughput and bytes received but still we observe that the results are not hundred percent. Our future work is to propose a new algorithm for congestion avoidance in congested network to improve the TCP environment.

ACKNOWLEDGMENT

I am deeply thankful to my HOD Prof Prashant Panse, Mahakal Institute of Technology, my brother Er. Preetam Suman, Research Scholar, IIITM Allahabad and all my friends whose help, stimulating suggestions and encouragement helped me in all the time for my research.

REFERENCES

[1] Stevens, W., TCP/IP Illustrated, Volume 1: The Procols Addison-Wesley, 1994.
[2] J.Postel,"Transmission Control Protocol", RFC 793, September, 1981.
[3] M.Allman RFC 2582: TCP Congestion Control.Technical report 1999.
[4] C.Siva Ram Murthy and BS.Manoj"Wireless Technology", 2008.
[5] M.Mathis, J Mahdan, S.Floyd, A.Roma now RFC 2018: TCP Selective Acknowledgment Options October 1996.
[6] K.Fall, S.Floyd" Simulation based comparison of Tahoe, Reno & SACK TCP", 1998.
[7] Renaud Buryeron, Bruno Hemon, and Lixia Zhang: "Experimentations with TCP Selective Acknowledgment", ACM SIGCOMM Computer Communication reviews, April 1988.
[8] L.S. Brakmo, L.L Peterson,"TCP Vegas: End to End congestion Avoidance on a Global Internet", IEEE journal on selected areas in communiction, vol.13, 1995.
[9] W.Stevens, TCP slow start, congestion avoidance, Fast retransmit & fast recovery algorithm, IETFRFC 2001, January 1997.