International Journal of Advanced Science and Research
ISSN: 2455-4227, Impact Factor: RJIF 5.12
www.allsciencejournal.com
Volume 2; Issue 4; July 2017; Page No. 119-122

Comparative analysis of TCP variants under DSDV routing protocol

Firdous Ul Rashid

Department of Computer Science & Engineering, Islamic University of Science and Technology, Awantipora Kashmir, Jammu and Kashmir, India

Abstract

A mobile ad hoc network (MANET) is generally defined as a network that has many free or autonomous nodes, often composed of mobile devices or other mobile pieces that can arrange themselves in various ways and operate without strict top-down network administration. In MANET each node acts as a router and these networks are scalable. To support connectivity between nodes MANET networks use different kinds of protocols such as AODV, DSR, and DSDV etc. MANETs is an IEEE 802.11 framework. It is an interconnected collection of wireless nodes where there is no networking infrastructure in the form of base stations, devices do not need to be within each other‘s communication range to communicate, the end- users devices also act as routers, nodes can enter and leave over time, data packets are forwarded by intermediate nodes to their final destination. The purpose of this paper is to analyze and compare the different TCP variants namely Reno, New Reno, Vegas under the routing protocol i.e. DSDV (Destination Sequenced Distance Vector).

Keywords: reno, new reno, vegas, MANET, DSDV

1. Introduction

Transmission Control Protocol (TCP), the mostly used transport protocol, performs well over wired networks. As much as wireless network is deployed, TCP should be modified to work for both wired and wireless networks. Since TCP is designed for congestion control in wired networks, it cannot clearly detect non-congestion related packet loss from wireless networks. TCP Congestion control plays the key role to ensure stability of the Internet along with fair and efficient allocation of the bandwidth. So, congestion control is currently a large area of research and concern in the network community. Many congestion control mechanisms are developed and refined by researcher aiming to overcome congestion. During the last decade, several congestion control mechanisms have been proposed to improve TCP congestion control. TCP is responsible for ensuring that a message is divided into the packets that IP manages and for reassembling the packets back into the complete message at the other end. In the OSI communication model, TCP is in layer 4, the Transport layer. TCP establishes a full duplex virtual connection between two endpoints.

TCP variants

TCP Reno

TCP Reno is the most widely adopted Internet TCP protocol. It employs four Congestion control Algorithms: slow start, congestion avoidance, fast retransmit, and fast recovery. When packet loss occurs in a congested link due to buffer overflow in the intermediate routers, either the sender receives three duplicate acknowledgments or the sender’s retransmission timeout (RTO timer expires). In case of three duplicate ACKs, the sender activates TCP fast retransmit and recovery algorithms and reduces its congestion window size to half. It then linearly increases congestion window, similar to the case of congestion avoidance. This increase in transmission rate is slower than in the case of slow start and helps relieve congestion. TCP Reno fast recovery algorithm improves TCP performance in case of a single packet loss within a window of data. However, performance of TCP Reno suffers in case of multiple packet losses within a window of data.

Disadvantages

RENO performs very well over TCP when the packet losses are small. But when we have multiple packet losses in one window then RENO does not perform too well and its performance is almost the same as Tahoe under conditions of high packet loss. Another problem is that if the window is very small when the loss occurs then we would never receive enough duplicate acknowledgements for a fast retransmit and we would have to wait for a coarse grained timeout. Thus is cannot effectively detect multiple packet losses.

New Reno

New Reno is a modification of TCP Reno. TCP New Reno enhances TCP throughput performance when multiple packets are dropped from a single window of data for TCP Reno connections that does not support the TCP SACK option. When multiple packets are dropped from a single window of data, the ACK for the retransmitted packet acknowledges some but not all of the packets transmitted before the fast retransmit. This is referred to as partial ACK. During fast recovery when a TCP sender receives partial ACK, the TCP sender concludes that the indicated packets was lost and retransmit that packet. The remaining three phases (slow start, congestion avoidance, and fast retransmit) are similar to TCP Reno. A problem occurs with New Reno when there are no packet losses but instead, packets are reordered by more than 3 packet sequence numbers. When this happens, New Reno mistakenly enters fast recovery, but when the reordered packet is delivered, ACK sequence-number progress occurs and from there until the end of fast recovery, every bit of sequence-number progress produces a duplicate and needless retransmission that is immediately ACKED.

Disadvantages

New-Reno suffers from the fact that it takes 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.

Vegas

Vegas is a TCP implementation which is a modification of RENO. It builds on the fact that proactive measure to encounter congestion is much more efficient than reactive ones. It tried to get around the problem of coarse grain timeouts by suggesting an algorithm which checks for timeouts at a very efficient schedule. Also it overcomes the problem of requiring enough duplicate acknowledgements to detect a packet loss, and it also suggests a modified slow start algorithm which prevents it from congesting the network.

The three major changes induced by Vegas are

- **New Re-Transmission Mechanism**: Vegas extend on the retransmission mechanism of RENO. It keeps track of when each segment was sent and it also calculates an estimate of the RTT by keeping track of how long it takes for the acknowledgment to get back.
- **Congestion avoidance:** TCP Vegas is different from all the other implementation in its behavior during congestion avoidance. It does not use the loss of segment to signal that there is congestion. It determines congestion by a decrease in sending rate as compared to the expected rate, as result of large queues building up in the routers.
- **Modified Slow-start:** TCP Vegas differs from the other algorithms during its slow-start phase. The reason for this modification is that when a connection first starts it has no idea of the available bandwidth and it is possible that during exponential increase it over shoots the bandwidth by a big amount and thus induces congestion. To this end Vegas increases exponentially only every other RTT, between that it calculates the actual sending throughput to the expected and when the difference goes above a certain threshold it exits slow start and enters the congestion avoidance phase.

Disadvantages

If there are enough buffer in the routers it means that Vegas congestion avoidance mechanism can function effectively a higher throughput and a faster response time result. As the load increase or the number or router buffer decrease, Vegas congestion avoidance mechanism is not as effective and Vegas start to behave more like Reno. Vegas is less aggressive in its use of router buffer than Reno because Vegas is limited. Finally Vegas congestion detection algorithm depends on the accurate value for Base RTT.

2. Destination sequenced distance vector

The destination sequenced distance vector routing protocol is a proactive routing protocol which is a modification of conventional Bellman-Ford routing algorithm. This protocol adds a new attribute, sequence number, to each route table entry at each node. Routing table is maintained at each node and with this table; node transmits the packets to other nodes in the network. This protocol was motivated for the use of data exchange along changing and arbitrary paths of interconnection which may not be close to any base station.

Advantages and Disadvantages

DSDV was one of the early algorithms available. It is quite suitable for creating ad hoc networks with small number of nodes. Since no formal specification of this algorithm is present there is no commercial implementation of this algorithm. DSDV guarantees for loop free path. DSDV requires a regular update of its routing tables, which uses up battery power and a small amount of bandwidth even when the network is idle. Whenever the topology of the network changes, a new sequence number is necessary before the network re-converges; thus, DSDV is not suitable for highly dynamic networks. (As in all distance-vector protocols, this does not perturb traffic in regions of the network that are not concerned by the topology change.

3. Simulation parameters

Table 1

Parameters	Values
Simulator used	NS2.3
No of Nodes	5,10,15,20
Simulation Time	190 sec
Traffic	FTP
TCP Variants	Reno, New Reno, Vegas
Packet Size	512
Window Size	15
Routing Protocol	DSDV
Queue Size	50

4. Performance Metrics

4.1 Throughput

It is defined as the ratio of the total number of bits received by the destination to the total simulation time. It is measured in bits per second/Kbps/Packets per second.

TP = received packets/simulation time (kbps)

4.2 End to End Delay

It is defined as the time taken for a packet to be transmitted across network from source to destination. The lower the delay the better is the performance.

4.3 Packet Delivery Ratio

It is defined as the ratio of the total number of packets received by the destination node to the number of packets send by the source node.

5. Simulation Results

In this section, we present the results of our ns-2 simulations of the three TCP variants namely Reno, New Reno and Vegas under AODV routing protocol. This wireless simulation has been done for 190 seconds.

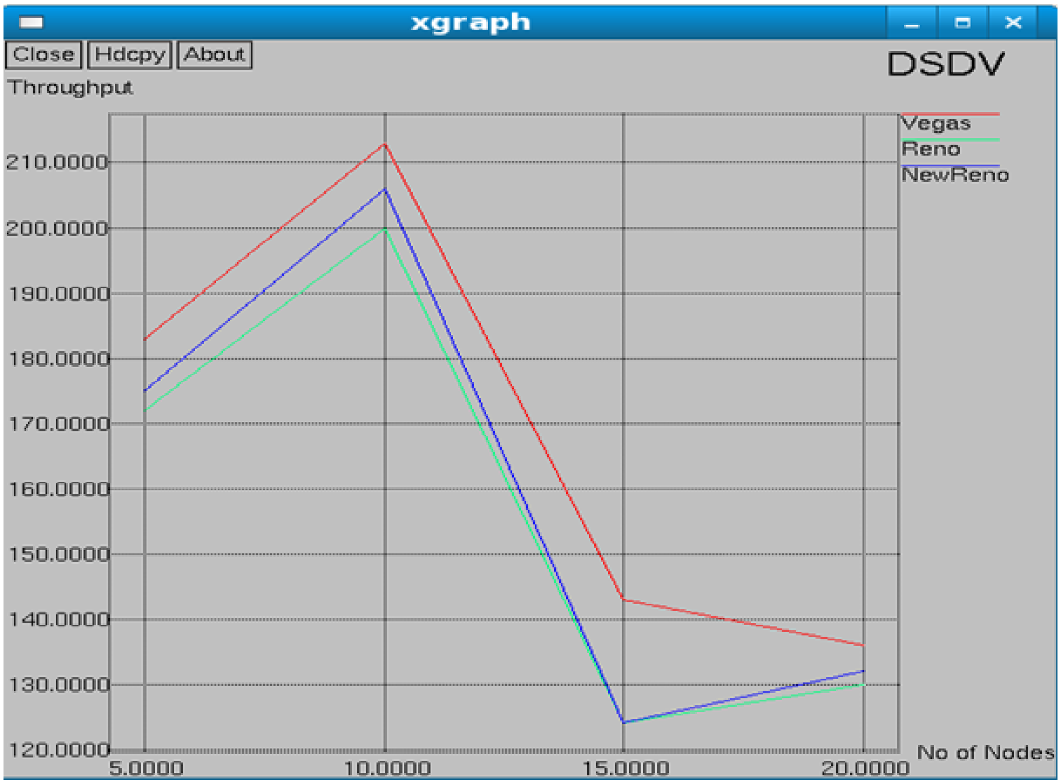

Fig 1: Comparison of TCP variants in terms of Throughput.

In fig1, we compare the TCP variants on the basis of Throughput and in the above graph we see that as the number of nodes increases the Throughput decreases in all the three TCP variants , but Vegas gives better Throughput as compared to Reno and New Reno. The reason behind that is , as we increase the node densities in a network the round trip time also increases due to which the Throughput decreases.

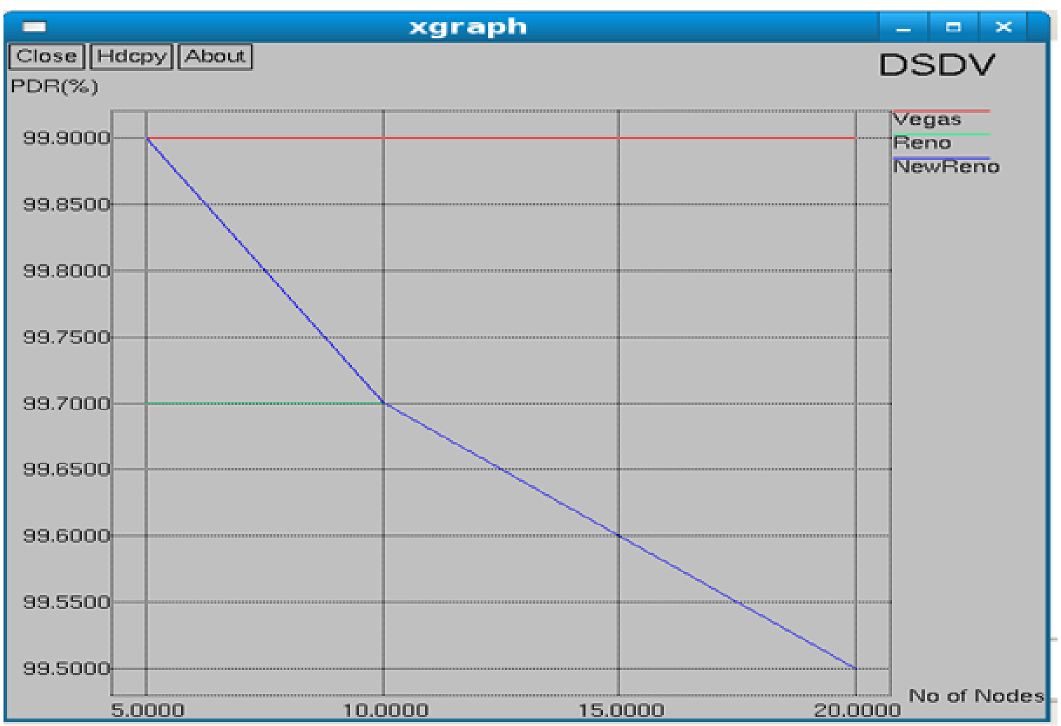

Fig 2: Comparison of TCP variants in terms of PDR

In fig 2, the packet delivery ratio of Vegas remains same as the node density increases and the PDR of TCP Reno and TCP New Reno decreases as the node density increases. Hence the performance of Vegas is better because the maximum numbers of packets are received by the destination node as compared to Reno and New Reno.

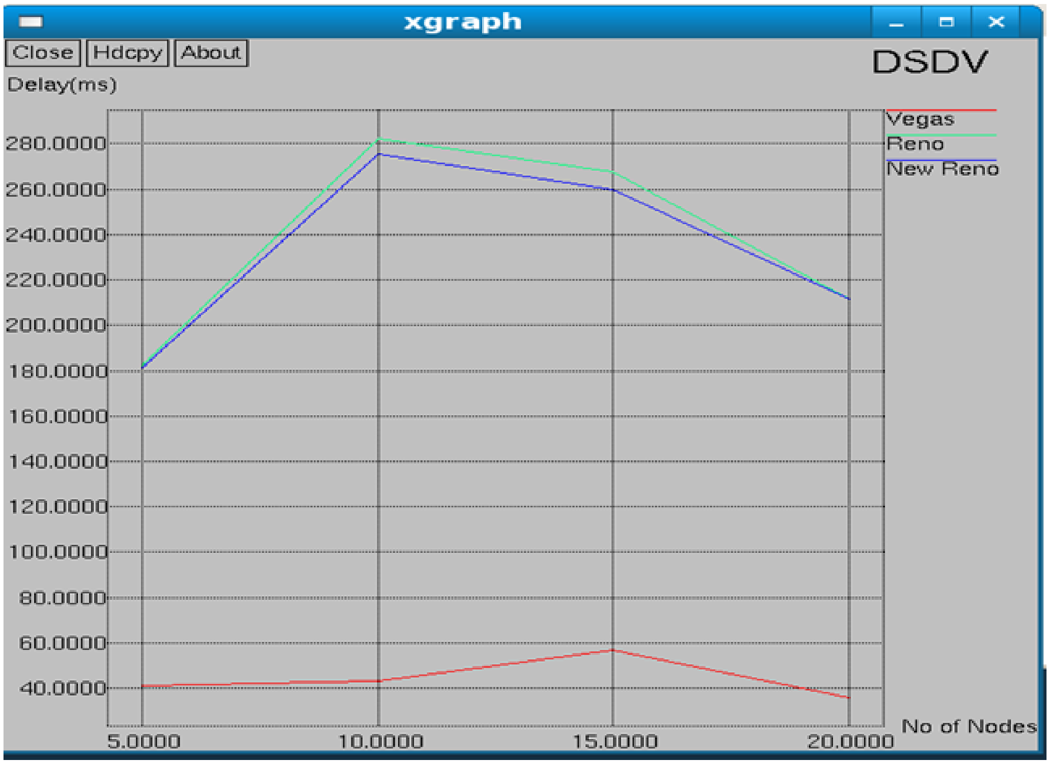

Fig 3: Comparison of TCP variants in terms of Delay

Fig 3 proves the End to End Delay of TCP variants namely Reno, New Reno and Vegas under different node densities. From the experimented results the TCP Vegas shows less delay than Reno and New Reno.

6. Conclusion

I have successfully evaluated the three TCP variants using NS2 simulation tool in the Mobile Ad hoc Networks. The results are more significant and comparable. From the simulation results, I conclude that TCP Vegas is much better than TCP Reno and New Reno because the TCP Vegas provides good Throughput than other two TCP variants and the packet delivery ratio of Vegas is better than Reno and New Reno.

7. Acknowledgment

I am very thankful to my friends Mr. Rafiq ,Mr Showkat for their guidance, support, stimulating suggestions and encouragement with the help of which I were able to complete this research work.

8. References

1. Jacobson V. Congestion Avoidance and Control. SIGCOMM Symposium no Communication Architecture and protocols. IJCN. 1988; 2.
2. Fall K, Floyd S. Simulation Based Comparison of Tahoe, Reno and SACK TCP. International Journal of Advances in Engineering & Technology, 2011.
3. Brakmo LS, Peterson LL. TCP Vegas: End to End Congestion Avoidance on a Global Internet, IEEE Journal on Selected Areas in Communication. 1995; 13:1465-1490.
4. Jehan M, Dr. Radhamani, KalaKumari T. VEGAS: Better Performance than other TCP congestion control algorithms on MANET's international journal of computer Networks IJCN. 2011; 3(2).
5. Vishnu Kumar Sharma, Dr. Sarita Singh. Bhadauria Performance Analysis on Mobile Agent Based Congestion Control using AODV routing protocol techniques with Hop by Hop algorithms for MANET" International Journal of Ad hoc, Sensor & Ubiquitous Computing IJASUC. 2012; 3(2).
6. Poonam Tomar, Shweta Yadav. Enhanced Reliable TCP for congestion control with corruption control in MANETs Journal of Global Research in Computer Science. 2012; 3(4).
7. Prof. Jain SA, Mr. Abhishek Bande, Mr. Gaurav Deshmukh, Mr. Yogesh Rade, Mr. Mahesh Sandhanshiv. An Improvement in Congestion Control Using Multipath Routing in Manet International Journal of Engineering Research and Applications IJERA. 2012; 2(3).
8. Mandakini. Tayade and Sanjeev Sharma Review of different TCP variants in ad hoc networks. IJEST ISSN: 0975-5462.