ISSN:0975-9646

Analyzing Different Performance Metrics on TCP Varients over Mobile Ad-Hoc Networks

Ms. Payal Sharma

Abstract-Wireless communication is an ever-developing field and the future holds many possibilities in this area. From the early radio and telephone to current devices such as mobile phones and laptops, accessing the global network has become the most essential and indispensable part of every one's lifestyle. Dominant means of supporting such communication capabilities is through the use of wireless LANs. As the deployment of wireless LANs increases around the globe, it is important to understand the different technologies and select the most appropriate one. A Mobile Ad hoc Network (MANET) is a dynamic multi-hop wireless network that is established by a group of mobile stations without necessarily using pre-existing infrastructure or centralized administration. It can be easily deployed which makes it very attractive for civilian and military applications. It is an infrastructure less network where self-configuring mobile nodes are connected by wireless links. Because of its decentralized property, these nodes relay on each other to store and forward packets. These are characterized by bandwidth constrained links, varying link qualities, and highly dynamic topologies.

In the proposed work, TCP sender side mechanisms and appropriate queue management algorithm to handle higher offered load, random losses and retransmission timeouts in high delay networks in such a way as to keep congestion window as high as possible, while keeping the congestion under control and keep retransmissions to minimal. The TCP proposed mechanisms are assessed against TCP RENO, New RENO, TCP VEGAS and Active queue management algorithm to see how they fare against congestion and higher offered load.

Keyword- TCP, RENO, VEGAS, ADHOC, AODV, MANET, DSR, WRP, IP, FTP, TELNET, HTTP.

I. INTRODUCTION TO MOBILE ADHOC NETWORK

A Mobile Ad Hoc Network (MANET) consists of a set of mobile hosts that carry out basic networking functions like packet forwarding, routing, and service discovery without the help of an established infrastructure. Nodes of an ad hoc network rely on one another in forwarding a packet to its destination, due to the limited range of each mobile host's wireless transmissions. An ad-hoc network uses no centralized administration. This ensures that the network will not cease functioning just because one of the mobile nodes moves out of the range of the others. Nodes should be able to enter and leave the network as they wish. Because of the limited transmitter range of the nodes, multiple hops are generally needed to reach other nodes. Every node in an ad hoc network must be willing to forward packets for other nodes. Thus, every node acts both as a host and as a router. The topology of ad hoc networks varies with time as nodes move, join or leave the network. This topological instability requires a routing protocol to run on each node to create and maintain routes among the nodes.

Ad hoc networks are also capable of handling topology changes and malfunctions in nodes. It is fixed through network reconfiguration. For instance, if a node leaves the network and causes link breakages, affected nodes can easily request new routes. Although there are incremental delays, the network continues to remain operational. Wireless ad hoc networks take advantage of the inherent nature of the wireless communication medium. In a wired network, the physical cabling is done a priori, restricting the connection topology of the nodes. Provided two mobile nodes are within transmission range of each other, this restriction is easily overcome within the wireless domain, forming an instantaneous communication link. Ad hoc networks are useful for the applications such as disaster recovery, automated battlefields, agriculture fields, security and vigilance, search and rescue, crowd control, conferences, meetings, and lectures where central or fixed infrastructure is not available.

There are many challenges in the creation of an ad hoc network: Heterogeneity, Routing challenges, wireless medium challenges, portability challenges, security, and scalability. MANETs are characterized by the mobility of nodes, which can move in any direction and at any speed that may lead to arbitrary topology and frequent partition in the network. This characteristic of the network makes the development of routing protocols as one of the most challenging issue.

In view of the necessity of developing efficient routing protocols, the present work focuses on comparative analysis of proactive(WRP) and reactive(AODV, DSR) routing protocols when traffic generator is FTP,TELNET or HTTP.

A. *TCP in the Internet Protocol Stack:* Figure 1 shows the structure of the Internet protocol stack, in which the TCP/IP is composed of the Network (IP) layer and Transport (TCP) layer. Each layer is responsible for a particular purpose which is to make various hosts to communicate with each other; hosts may be computers, or processes within a computer. (IPS) Internet protocol stack is redesigned from the formerly used OSI reference model.

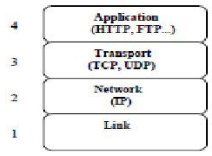

Figure 1: Internet Protocol Stack

The application layer is liable for the production and consumption of the user's data which passes through each layer of the stack and is transferred transversely the network. The transport layer is responsible for the end-to-end transmission of the data formed by the application layer. The network layer is mainly work with routing of packets between sender and receiver hosts. The Network layer supports different routing protocol like Dynamic Source Routing Protocol (DSR), Temporally Ordered Routing Algorithm (TORA). The Data-link layer or network interface layer specifies the mechanism of how the packets of the network layer are transported over the physical medium between two nodes. The data link layer deals with physical transmission details such as frame size, synchronization, etc.

i. *Connection Setup:* In TCP, both the hosts (sender and receiver) want to communicate with each other for a certain amount of time, they handshake with each other. Handshaking consists of three phases. Connections are established in TCP by means of the three-way handshake. To establish a connection, one side, say, the server passively waits for an incoming connection by executing the LISTEN and ACCEPTS primitives, either specifying a specific source or nobody in particular. The other side, say, the client, executes a CONNECT primitive, specifying the IP address and port to which it wants to connect, the maximum TCP segment size it is willing to accept, and optionally some user data (e.g., a password). The CONNECT primitive sends a TCP segment with the SYN bit on and ACK bit off and waits for a response. When this segment arrives at the destination, the TCP entity checks to see that if there is a process that has done a LISTEN on the port given in the Destination port field. If not, it sends a reply with the RST bit on to reject the connection. If some process is listening to the port, that process is given the incoming TCP segment. It can then either accept or reject the connection. If it accepts, an acknowledgement segment is sent back.

ii. *Connection Release:* Although TCP connections are full duplex, to understand how connections are released it is best to think of them as a pair of simplex connections. Each simplex connection is released independently of its sibling. To release a connection, either party can send a TCP segment with the FIN bit set, which means that it has no more data to transmit. When the FIN is acknowledged, the direction gets shut down for new data. Data may continue to flow indefinitely in the other direction, however. When both directions have been shut down, the connection is released. There is, in fact, no essential difference between the two hosts releasing sequentially or simultaneously. To avoid the two-army problem, timers are used. If a response to a FIN is not forthcoming within two maximum packet lifetimes, the sender of the FIN releases the connection. The other side will eventually notice that nobody seems to be listening to it anymore and will time out as well.

B. *TCP Congestion Control:* When the load offered to any network is more than it can handle, congestion builds up. Congestion can be dealt with by employing a principle borrowed from physics: the law of conservation of packets. The idea is to refrain from injecting a new packet into the network until an old one leaves (i.e., is delivered). TCP attempts to achieve this goal by dynamically manipulating the window size. The first step in managing congestion is detecting it. In the old days, detection of congestion was difficult. A timeout caused by a lost packet could have been caused by either (1) noise on a transmission line or (2) packet discard at a congested router. Telling the difference was difficult. Now a day, packet loss due to transmission errors is relatively rare because most long-haul trunks are fiber. Consequently, most transmission timeouts on the Internet are due to congestion. All the Internet TCP algorithms assume that timeouts are caused by congestion and monitor timeouts for signs of trouble the way miners watch their canaries. When a connection is established, a suitable window size has to be chosen. The receiver can specify a window based on its buffer size. If the sender sticks to this window size, problems will not occur due to buffer overflow at the receiving end, but they may still occur due to internal congestion within the network.

C. Congestion Avoidance Algorithm

Congestion avoidance is a way to deal with lost packets. The assumption of the algorithm is that packet loss caused by damage is very small (much less than 1%), therefore the loss of a packet signals congestion somewhere in the network between the source and destination. There are two indications of packet loss: a timeout occurring and the receipt of duplicate ACKs. If we are using a timeout as an indication of congestion, we can see the need for a good RTT algorithm, such as that described .Congestion avoidance and slow start are independent algorithms with different objectives. But when congestion occurs we want to slow down the transmission rate of packets into the network, and then invoke slow start to get things going again. Congestion avoidance and slow start require that two variables be maintained for each connection: a congestion window (CWND) and a slow start threshold size (SSTHRESH). The combined algorithm operates as follows:

Initialization for a given connection sets CWND and SSTHRESH.

The TCP output routine never sends more than the minimum of CWND and the receiver's advertised window.

Congestion avoidance is flow control imposed by the sender, while the advertised window is flow control imposed by the receiver. The former is based on the sender's

assessment of perceived network congestion; the latter is related to the amount of available buffer space at the receiver for this connection.

When congestion occurs (indicated by a timeout or the reception of duplicate ACKs), one-half of the current window size is saved in SSTHRESH. Additionally, if the congestion is indicated by a timeout, cwnd is set to one segment that is slow start.

When new data is acknowledged by the other end, we increase CWND, but the way it increases depends on whether we are performing slow start or congestion avoidance.

If CWND is less than or equal to SSTHRESH, we're doing slow start; otherwise we are doing congestion avoidance. Slow start continues until we are halfway to where we were when congestion and then congestion avoidance takes over.

II. LITERATURE REVIEW

P. Arivubrakan et. al. [1] discussed the transmission range in terms of distance.. From experimental analysis, this was concluded that DSR has maximum throughput, high packet delivery ratio, loss of packet is less and end to end delay is low compared to the AODV routing protocol.

Nur Idawati et. al. [2] did the evaluation about improvement in Quality of Service (QoS) for AODV routing protocol . This work highlights the combination of both maximum delay extension and minimum bandwidth extension metrics.

Author in [3] study the impact of node mobility and radio channels on link statistics in mobile ad hoc networks (MANETs) and how to achieve desired network performance. They find that the impacting factors on residual link lifetime are in the decreasing order of node speed, transmission range, node-pair distance.

Author in [4] discussed that Mobile Ad Hoc Networks (MANETs) are generating a lot of interests due to 3G and 4G activities. In this, the author concentrates on routing, which is a challenging task and has seen a huge number of different strategies proposed, each claiming to provide an improvement over other strategies?

S. Rajeswari et. al. [5] discussed that QoS improvement has been a subject of intensive discussion. In this, the author noticed that using RED has greatly improved all the performance measures especially with FIFO. The reason is that RED monitors the average queue size and randomly drops packets when congestion is detected.

Visvasuresh Victor in [6] present a mathematical model for anovel TCP congestion control approach called Receiver-Window Modification (RWM).

Younghwan et. al. In [7] discussed in this the relation between distance and hop count first. Then, based on the relation, the required hop count between two given L1MNs is estimated. With the distribution of distance between pairs of MNs, this paper also suggests the optimal transmission power of MNs, which can guarantee delay constraints of a requested ratio of connections in an entire network.

Chen, I. in [8] analyse the TCP performance in multi hop wireless networks based on the injected traffic and the control traffic.

Author in [9] propose a receiver-aided mechanism in which the TCP receiver monitors the contention state of the connection and accordingly informs the TCP sender about it via ACK mechanism. TCP receiver uses end-to-end delay as contention criteria.

Research work present in [10] propose a mechanism called DDLRP (Detecting and Differentiating the Loss of Retransmitted Packets) which detects and differentiates the loss of retransmitted packets and reacts by retransmitting the packet without waiting for the retransmission timeout. DDLRP consists of two schemes, namely, Retransmission Loss Detection and Retransmission Loss Differentiation.

Y.C. Hu et. al. [11] surveys the secure wireless ad hoc routing. This paper reviews routing attacks on ad hoc networks, security in routing Protocol and discussed current approaches for establishing cryptographic keys in ad hoc networks and describes the state of research in secure ad hoc routing protocols and its research challenges.

Author in [12] given details about security threats like impersonation, modification and fabrication attacks against ad hoc routing protocol specifically for AODV and DSR and purposed ARAN a secure routing protocol based on certificates and successfully defeats all identified attacks.

III. PROBLEM STATEMENT

In this research work, TCP sender side mechanisms and appropriate queue management algorithm to handle higher offered load, random losses and retransmission timeouts in high delay networks in such a way as to keep congestion window as high as possible, while keeping the congestion under control and keep retransmissions to minimal. The TCP proposed mechanisms are assessed against TCP RENO, New RENO, TCP VEGAS and Active queue management algorithm to see how they fare against congestion and higher offered load. Ns2 simulator is selected as the simulation tool because of the ease of use of the graphical interface provided and extensive support of TCP. Also the free license availability for research purpose encouraged us to select ns2 simulator.

Transport Control Protocol (TCP) is a connection oriented protocol of the transport layer. It provides features like reliability, flow control and congestion control. As TCP was designed for wired networks it considers that all packet loss in the network is due to congestion. Wireless medium is more exposed to transmission errors and sudden topological changes. So when we adapt TCP to ad hoc networks It misinterprets the packet losses due to link failure as packet losses due to congestion and in the instance of a timeout, backing-off its retransmission timeout (RTO). This results in unnecessary reduction of transmission rate because of which throughput of the whole network degrades. Due to high error rates and connectivity characteristics of wireless links, TCP reacts to packet loss as it would in wired environment. It

drops the transmission window size before retransmitting packets and initiates congestion control or avoidance mechanism such as slow start and resets its transmission timer. TCP-Tahoe, TCP-Reno, TCP-New Reno, TCP Sack, TCP-Vegas and TCP- New Jersey are some of the most important variants of TCP. Depending on the scenario selection of TCP variant has to be done.

The four algorithms, Slow Start, Congestion Avoidance, Fast Retransmit and Fast Recovery are described below.

A. *Slow Start:* Slow Start, a requirement for TCP software implementations is a mechanism used by the sender to control the transmission rate, otherwise known as sender based flow control. This is accomplished through the return rate of acknowledgements from the receiver. When the TCP connection first begins, then the Slow Start algorithm initializes a congestion window to one segment, which is the maximum segment size (MSS) initialized by the receiver during the connection establishment phase. When acknowledgements are returned by the receiver, the congestion window increases by one segment for each acknowledgement returned. Thus, the sender can transmit the minimum of the congestion window and the advertised window of the receiver, which is simply called the transmission window. At some point the congestion window may become too large for the network or network conditions may change such that packets may be dropped. Packets lost will trigger a timeout at the sender. When this happens, the sender goes into congestion avoidance mode as described in the next section.

B. Congestion Avoidance: During the initial data transfer phase of a TCP connection the Slow Start algorithm is used. However, there may be a point during Slow Start that the network is forced to drop one or more packets due to overload or congestion. If this happens, Congestion Avoidance is used to slow the transmission rate. In the Congestion Avoidance algorithm a retransmission timer expiring or the reception of duplicate ACKs can implicitly signal the sender that a network congestion situation is occurring. The sender immediately sets its transmission window to one half of the current window size (the minimum of the congestion window and the receiver's advertised window size), but to at least two segments. If congestion was indicated by a timeout, the congestion window is reset to one segment, which automatically puts the sender into Slow Start mode. If congestion was indicated by duplicate ACKs, the Fast Retransmit and Fast Recovery algorithms are invoked.

C. *Fast Retransmit:* When a duplicate ACK is received, the sender does not know if it is because a TCP segment was lost or simply that a segment was delayed and received out of order at the receiver. If the receiver can re-order segments, it should not be long before the receiver sends the latest expected acknowledgement. Typically no more than one or two duplicate ACKs should be received when simple out of order conditions exist. If however more than two duplicate ACKs are received by the sender, it is a strong indication that at least one segment has been lost. The TCP sender will assume enough time has lapsed for all segments to be properly re-ordered by the fact that the receiver had enough time to send three duplicate ACKs. When three or more duplicate ACKs are received, the sender does not even wait for a retransmission timer to expire before retransmitting the segment.

D. *Fast Recovery:* Since the Fast Retransmit algorithm is used when duplicate ACKs are being received, the TCP sender has implicit knowledge that there is data still flowing to the receiver. The reason is because duplicate ACKs can only be generated when a segment is received. This is a strong indication that serious network congestion may not exist and that the lost segment was a rare event. So instead of reducing the flow of data abruptly by going all the way into Slow Start, the sender only enters Congestion Avoidance mode. Rather than start at a window of one segment as in Slow Start mode, the sender resumes transmission with a larger window, incrementing as if in Congestion Avoidance mode. This allows for higher throughput under the condition of only moderate congestion.

IV. TCP VARIANTS

A. *TCP TAHOE:* TAHOE refers to the TCP congestion control algorithm. TCP is based on a principle of conservation of packets, i.e. 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. TCP implements this principle by using the acknowledgements to clock outgoing packets because an acknowledgement means that a packet was taken off the wire by the receiver. It also maintains a congestion window CWD to reflect the network capacity. Tahoe suggests that whenever a TCP connection starts or restarts after a packet loss it should go through a procedure called slow-start. The reason for this procedure is that an initial burst might overwhelm the network and the connection might never get started.

B. *TCP RENO:* This RENO retains the basic principle of Tahoe, such as slow starts and the coarse grain retransmit timer. However it adds some intelligence over it so that lost packets are detected earlier and the pipeline is not emptied every time a packet is lost. RENO requires that we receive immediate acknowledgement whenever a segment is received. The logic behind this is that whenever we receive a duplicate acknowledgment, then his duplicate acknowledgment could have been received if the next segment in sequence expected, has been delayed in the network and the segments reached there out of order or else that the packet is lost. If we receive a number of duplicate acknowledgements then that means that sufficient time have passed and even if the segment had taken a longer path, it should have gotten to the receiver by now. There is a very high probability that it was lost. So Reno suggests Fast Retransmit.

C. *TCP VEGAS:* Bandwidth Estimation scheme used by TCP Vegas is more efficient than other TCP variants. This scheme makes bandwidth estimation by using the difference between the expected flow rates and the actual flow rates. It extends TCP-RENO by modifying its Congestion Avoidance mechanism. Like TCP-Reno it uses Slow Start and Fast Retransmission. TCP-VEGAS use its Congestion Avoidance mechanism in order to avoid packet loss by decreasing its CWND as soon as it detects congestion in the network.

D. *TCP SACK:* TCP with 'Selective Acknowledgments' is an extension of TCP RENO and it works around the problems face by TCPRENO and TCP New-Reno, namely detection of multiple lost packets, and re-transmission of more than one lost packet per RTT. SACK retains the Slow-Start and Fast Re-Transmit parts of RENO. It also has the coarse grained timeout of Tahoe to fall back on, in case a packet loss is not detected by the modified algorithm.

3.5 Simulation Environment

This simulation process considered a wireless network of various network sizes consisting of 40, 60, 80 and 100 nodes which are placed within a 1500m x 1500m area. FTP traffic is generated among the nodes. The simulation runs for 150 Seconds. Table 1 shows the important simulation parameters used in the simulation process.

Table 1: Important Simulation Parameters

Parameter	Value
Simulation time	150 Sec
Simulation area	1500m x 1500m
Antenna	Omni antenna
No. of nodes	40, 60, 80, 100
TCP Varients	RENO, TAHOE, SACK, VEGAS
Interface Queue Type	DropTrail-PriQueue, RED
Packet size	512 Bytes
Max queue length	50
Traffic	FTP
Routing protocol	DSDV, OLSR
Mobilty Model	Random Waypoint Model

Following performance metrics are used to evaluate and analyze the performance of various routing protocols:

Packet Delivery Ratio: The ratio of data packets delivered to the destinations to those generated by the sources

Average End to End Delay: The average delay a data packet takes to travel from the source to the destination node.

Throughput: Number of bits delivered successfully per second to the destination. It is the measure of effectiveness.

Routing Message Overhead: It is calculated as total number of control packets transmitted. The increase in routing message overhead reduces the performance of the ad hoc network.

V. Results And Discussions

Simulations are performed for different routing layer protocols in a multi-hop ad hoc network environment. The impact of network density with different TCP variants and offered load on the performance of DSDV and OLSR routing protocols is shown with the help of graphs in terms of packet delivery ratio, throughput, end-to-end delay and routing overhead A. TCP- TAHOE: Figure 2 shows the throughput when the network density is varied between 40 to 100. The throughput of DSDV is higher than OLSR under TCP-TAHOE without RED and with RED. With Active Queue Management technique-RED, simulation results shows improvement in throughput of these routing protocols under congested network environment.

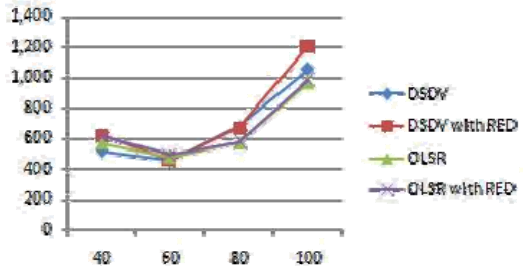

Figure 2: Throughput for DSDV and OLSR with and without RED

Figure 3 shows the packet delivery ratio when the network density is varied between 40 to 100. The packet delivery ratio of OLSR is higher than DSDV under TCP-TAHOE without RED and with RED. With Active Queue Management technique-RED, simulation results shows improvement in packet delivery ratio under congested network environment.

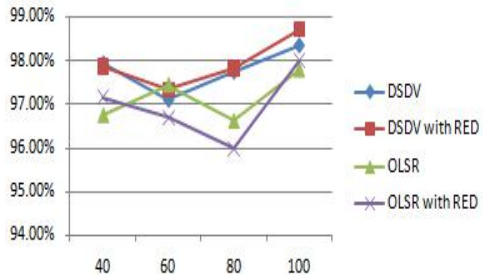

Figure 3: Packet Delivery Ratio for DSDV and OLSR with and without RED

After analysis of simulation graph for routing protocols under TCP- TAHOE with Active Queue Management technique RED, there is improvement in quality of service of routing protocols under congested environment. So the performance of DSDV and OLSR is evaluated for other TCP variants-TCP-RENO, TCP-SACK and TCP-VEGAS.

B. TCP-RENO: Figure 4 shows the throughput when the network density is varied. The throughput of DSDV is higher than OLSR under TCP-RENO which means DSDV is better than OLSR.

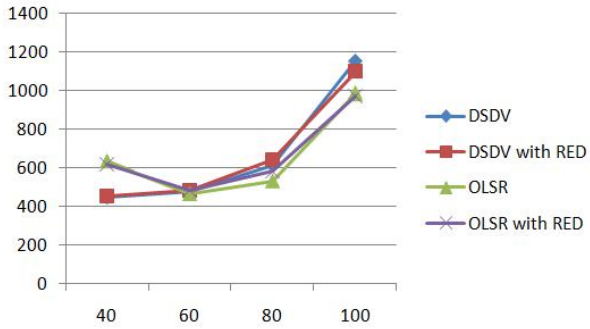

Figure 4: Throughput for DSDV and OLSR with and without RED

Figure 5 shows the Packet Delivery ratio when the network density is varied. The packet delivery ratio of DSDV is higher than OLSR.

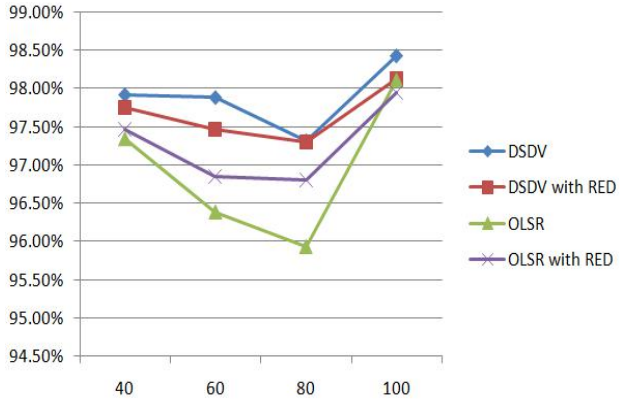

Figure 5: Packet Delivery Ratio for DSDV and OLSR with and without RED

C. TCP-SACK: Figure 6 shows the simulation result for throughput of DSDV and OLSR routing protocols under TCP-SACK.

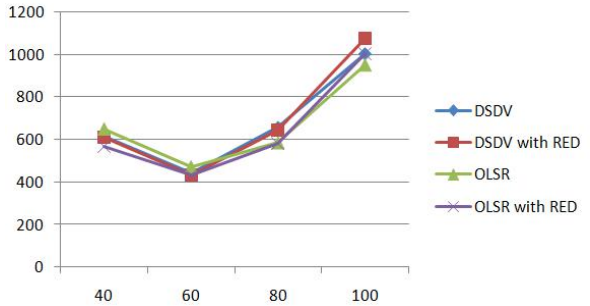

Figure 6: Throughput for DSDV and OLSR with and without RED

Figure 7 shows the simulation result for Packet Delivery Ratio of DSDV and OLSR routing protocols under TCP-SACK.

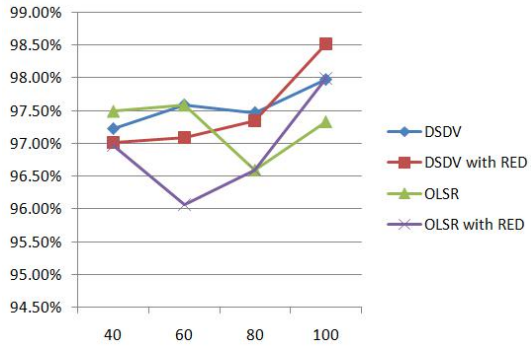

Figure 7: Packet Delivery Ratio for DSDV and OLSR with and without RED

D. TCP-VEGAS: Figure 8 shows the simulation result for Throughput of DSDV and OLSR routing protocols under TCP-Vegas.

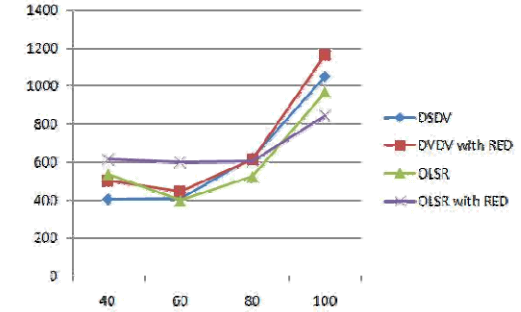

Figure 8: Throughput for DSDV and OLSR with and without RED

Figure 9 shows the simulation result for Packet Delivery Ratio of DSDV and OLSR routing protocols under TCP-Vegas.

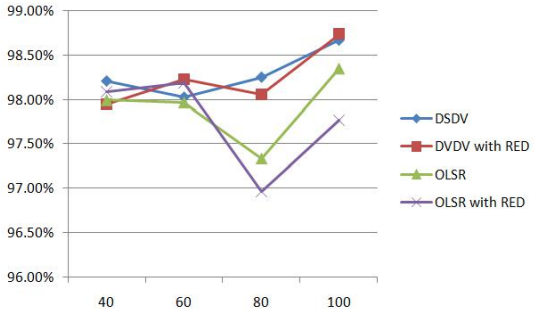

Figure 9: Packet Delivery Ratio for DSDV and OLSR with and without RED

VI. Conclusion And Future Scope

In this paper, the effect of Random Early Detection (RED) active queue management technique with different TCP variant – TCP-Tahoe, TCP-Reno, TCP-SACK and TCP-Vegas under varying congested network density is examined on to check the improvement and performance of Destination Sequence Distance Vector (DSDV) and Optimized Link State Routing (OLSR) under the FTP traffic. From the

simulation results it is observed that firstly different TCP variants (TCP-Tahoe, TCP-Reno, TCP-Sack and TCP-Vegas) with DropTail queue were run without active queue management technique to evaluate the performance of routing protocols. Then all this was repeated using RED active queue management technique. So the overall performance of DSDV is much better. But in case of congestion avoidance algorithm, TCP-VEGAS with RED improve the performance of routing protocol are better than other TCP variant.

REFERENCES

[1] P. Arivubrakan and V.R. Sarma Dhulipala (2012), "QoS Enhancement by varying Transmission Range in Wireless Ad-hoc Networks", International Journal of Computer Applications (0975 – 8887) Volume: 37, No.9, pp:1-4,January 2012.

[2] Nur Idawati Md Enzai, Farhat Anwar and Omer Mahmoud (2008), "Evaluation Study of QoS-Enabled AODV", Proceedings of the International Conference on Computer and Communication Engineering, Kuala Lumpur, Malaysia, pp: 1254-1259, 2008.

[3] . Ming Zhao and Wenye Wang (2007), "The Impacts of Radio Channels and Node Mobilityon Link Statistics in Mobile Ad Hoc Networks", IEEE GLOBECOM 2007 proceedings, pp: 1206-1210, 2007.

[4] Shima Mohseni, Rosilah Hassan, Ahmed Patel, and Rozilawati Razali (2010), "Comparative Review Study of Reactive and Proactive Routing Protocols in MANETs", 4th IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST 2010), pp:304-309, 2010.

[5] S.Rajeswari, Y.Venkataramani, "Congestion Control and QOS Improvement forAEERG protocol in MANET",International Journal on AdHoc Networking Systems (IJANS) Vol. 2, No. 1, pp:13-21, 2012.

[6] . Visvasuresh Victor Govindaswamy, Gergely Záruba and G.Balasekaran (2006), "Receiver-Window Modified Random Early Detection(RED-RWM) Active Queue Management Scheme:Modeling and Analysis", IEEE ICC proceedings, pp:158-163, 2006 .

[7] Younghwan Yoot and Dharma P. Agrawali (2007), "Optimal Transmission Range with Delay Constraints for Homogeneous MANETs",IEEE conference, 2007.

[8] Chen, I. Marsic, R. Miller, "Issues and Improvements in TCP Performance over Multihop Wireless Networks", IEEE Sarnoff Symposium, 2008.

[9] C. Chen, H. Wang, XinWang, M. Li, A.O. Lim, "A Novel Receiver-aided Scheme for Improv-ing TCP Performance in Multihop Wireless Networks", in Proc. of Int. Conference on Communications and Mobile Computing, pp:272–277, 2009.

[10] S. Prasanthi, S. Chung, C. Ahn, "An Enhanced TCP Mechanism for Detecting and Differen-tiating the Loss of Retransmissions over Wireless Networks", in Proc. of Int. Conference on Advanced Information Networking and Applications, pp: 54– 61, 2011.

[11] . Y.C. Hu and A. Perrig, "A Survey of Secure Wireless Ad Hoc Routing,", IEEE Security and Privacy, vol 2, no 3, pp:28–39, 2004.

[12] . K. Sanzgiri, B. Dahill, B.N. Levine, C. Shields and E.M. Royer, "A Secure Routing Protocol for Ad hoc Networks", Proc. of IEEE International Conf. Network Protocols, pp:78-87,2002.