www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume - 3 Issue -9 *September*, **2014 Page No. 8170-8174**

A Survey on Routing Protocols Using TCP Variants over MANETs

Manpreet Kaur[1], Dr. Sandeep Singh Kang[2]

[1] M.tech Scholar CGC, College of Engineering,
Landran Road, Mohali, India
man.bal81@gmail.com

[2]HOD (CSE), CGC, College of Engineering,
Landran Road, Mohali, India
sskang4u1@rediffmail.com

Abstract: ***Ad hoc networks are characterized by multi-hop wireless connectivity, frequently changing network topology and the need for efficient dynamic routing protocols plays an important role. Comparison of the performance of two prominent on-demand routing protocols for mobile ad hoc networks: Dynamic Sequence Distance Vector (DSDV) Routing Protocol, Optimized Link State Routing (OLSR) Protocol. A detailed simulation model with MAC and physical layer models is used to study the interlayer interactions and their performance implications. This paper demonstrate that even though OLSR and DSDV share similar on-demand behaviour, the differences in the protocol mechanisms can lead to significant performance differentials. Overview of two on demand routing protocols DSDV and OLSR based on packet delivery ratio, normalized routing load, normalized MAC load, average end to end delay by varying the number of sources, speed and pause time. Reliable transport protocols such as TCP are tuned to perform well in traditional networks where packet losses occur mostly because of congestion. However, networks with wireless and other lossy links also suffer from significant losses due to bit errors and handoffs. TCP responds to all losses by invoking congestion control and avoidance algorithms, resulting in degraded end-to-end performance in wireless and lossy systems. In this Paper, comparison of several schemes designed to improve the performance of TCP in such networks is included.***

Keywords: MANETs**,** TCP variants, Routing Protocols, TCP Congestion Control.

1. Introduction

Wireless nodes in network communicate to other nodes through wireless links within specific area directly or indirectly with the help of intermediate node. In wireless networks, nodes transmit information through electromagnetic propagation over the air. The signal transmitted by a node can only be received by nodes that are located within a specific distance from the transmitting node. The concept of dynamic mobility is introduced in this network because nodes are moving from one place to another place, within this network any node can join the network and can leave the network at any time. Nodes can be the form of systems or devices i.e. mobile phone, laptop, personal digital assistance, MP3 player and personal computer that are participating in the network and are mobile. Security and immediate reply of different types of nodes (end to end nodes, intermediate nodes and wireless antenna) is the main concern in AD HOC networks [1]. A mobile ad-hoc network (MANET) is a self-configuring infrastructure less network of mobile devices connected by wireless links. Ad-hoc is Latin and means "for this purpose". Each device in a MANET is free to move independently in any direction, and will therefore change its links to other devices frequently. Each must forward traffic unrelated to its own use, and therefore be a router. The primary challenge in building a MANET is equipping each device to continuously maintain the information required to properly route traffic. Such networks may operate by themselves or may be connected to the larger Internet. One of the important characteristics of a MANET node is the neighbor discovery for the data reception and transmission. It has flexible network architecture and variable routing paths to provide communication in case of the limited wireless connectivity range and resource constraints. The mobile ad hoc network has the following typical characteristics:

- **Unreliability of wireless links between nodes**. Because of the limited energy supply for the wireless nodes and the mobility of the nodes, the wireless links between mobile nodes in the ad hoc network are not consistent for the communication participants [2].
- **Bandwidth-Constrained, variable capacity links:** Wireless links will continue to have significantly lower capacity than their hardwired counterparts. In addition, the realized throughput of wireless communications (after accounting for the effects of multiple access, fading, noise, and interference conditions, etc.) is often much less than a radio's maximum transmission rate. One effect of the relatively low to moderate link capacities is that congestion is typically very common and the mobile users will demand similar services like the ones served by its fixed counterpart. These demands will continue to increase as multimedia computing and collaborative networking applications rise.
- **Multi-hop communications:** Due to signal propagation characteristics of wireless transceivers, ad hoc networks require the support of multi hop communications; that is mobile nodes that cannot reach the destination node directly will need to relay their messages through other nodes [2].

- **Constantly changing topology:** Due to the continuous motion of nodes, the topology of the mobile ad hoc network changes constantly: the nodes can continuously move into and out of the radio range of the other nodes in the ad hoc network, and the routing information will be changing all the time because of the movement of the node.

1.2. Routing in MANET

The task of routing in mobile ad hoc network is non-trivial since host mobility and changes in node activity status cause frequent unpredictable topological changes. The simplest approach to routing in a dynamic topology would be flooding the network with a packet to be sent, with the hope that it would eventually reach the destination [6]. However, this is extremely inefficient. Routing protocols for ad hoc networks assume a rate of topology change not high enough to make flooding the only alternative and not low enough to make conventional routing protocols effective. A routing protocol for ad hoc networks must be distributed, since in view of the dynamic topology no centralized point of control is possible. It should generate routes quickly so that they can be used before topology changes. Also, it must be bandwidth efficient, power conserving, and have minimal control overhead [7].

The design of a routing protocol is challenging due to the unique characteristics of Adhoc network, including resource scarcity or the unreliability of the wireless medium. There are various ways to classify routing protocols.

Routing protocols are responsible for identifying or discovering routes from a source or sender to the intended receiver. This route discovery process can also be used to distinguish between different types of routing protocols [4]. Reactive protocols discover routes on-demand that is, whenever a source wants to send data to a receiver and does not already have a route established.

While reactive route discovery incurs delays before actual data transmission can occur, proactive routing protocols establish routes before they are actually needed. This category of protocols is also often described as table-driven, because local forwarding decisions are based on the contents of a routing table that contains a list of destinations combined with one or more next-hop neighbors that lead toward these destinations and costs associated with each next hop option.

While table-driven protocols eliminate the route discovery delays, they may be overly aggressive in that routes are established that may never be needed. Further, the time interval between route discovery and actual use of the route can be very large; potentially leading to outdated routes (e.g., a link along the route may have broken in the meantime).

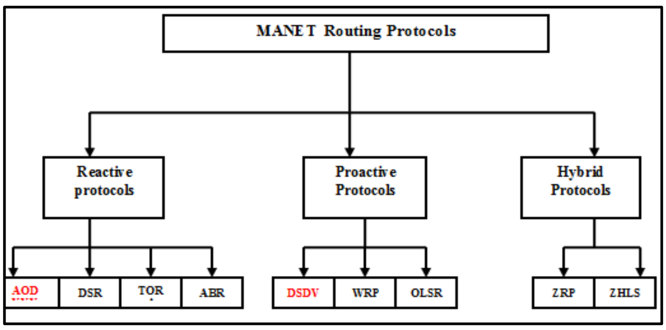

Fig 1: Classification of Ad Hoc Network Routing Protocols

1.3. Introduction to TCP

Transmission Control Protocol (TCP) is a connection oriented point-to-point protocol. It is a means for building a reliable communications stream on the top of the unreliable Internet Protocol (IP). TCP is the protocol that supports nearly all Internet applications. TCP is used by a large number of IP applications, such as email, Web services, and TELNET. As a connection-oriented protocol, TCP ensures that data is transferred reliably from a source to a destination [4].

Reliability in transmission involves the use of some form of handshake between the sender and receiver. Also, sequence numbers can be used to ensure in-sequence delivery of segments and help to identify lost or corrupted segments. Retransmission can be used to resend lost or corrupted segments. Hence, a retransmission timer is needed to determine when to initiate a resend. For TCP, an adaptive retransmission mechanism is employed to accommodate the varying delays encountered in the Internet environment. The timeout parameter is adjusted accordingly by monitoring the delay experienced on each connection.

TCP also responsible for managing network buffer overflows. TCP is transparent to the intermediate mobile nodes, the sender has to indirectly figure out network buffer overflows by keeping a timer that estimates the round-trip time (RTT) for TCP segments. If it does not receive an ACK packet before its timer expires, a sender will assume that the packet was lost owing to network congestion and will retransmit the packet.

1.4. TCP Congestion Control

When the load offered to any network is more than it can handle, congestion builds up. The Internet is no exception. 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. TCP attempts to achieve this goal by dynamically manipulating the window size. Figure 2 illustrates how TCP manage network congestion. TCP maintains two variables that is a congestion window and a slow-start threshold [9]. The congestion window determines the number of segments that is transmitted within an RTT. At the start of a TCP session, the congestion window is set to 1, and the transmitter sends only one segment and waits for an acknowledgment. When an ACK is received, the congestion window is doubled, and two segments are transmitted at a time. This process of doubling the Congestion window continues until it reaches the maximum

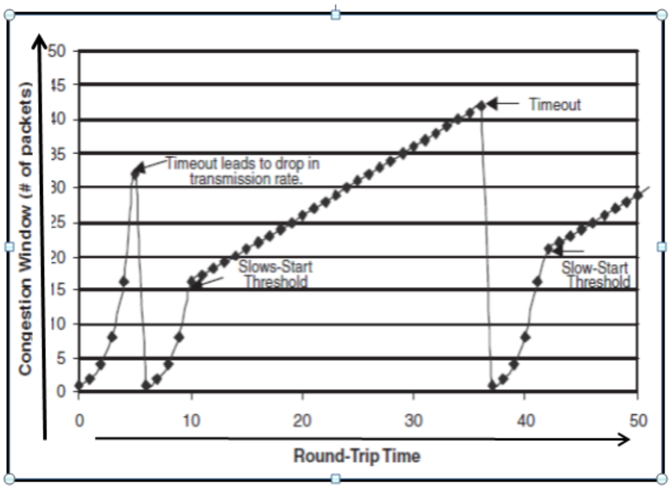

Fig 2: TCP congestion control

indicated by the advertised window size or until the sender fails to get an acknowledgment before the timer expires. At this point, TCP infers that the network is congested and begins the recovery process by dropping the congestion window back to one segment.

Resetting the congestion window to one segment allows the system to clear all packets in transit. Now, if a retransmission also fails, the TCP sender will also exponentially back off its retransmission time, providing more time for the system to clear the congestion. If transmission is successful after restart, the process of doubling the congestion window size after every transmission continues until the contention window size reaches half the size at which it detected the previous congestion. This is called the slow-start threshold. Once at this threshold, the congestion window is increased only linearly by one segment size at a time in what is called the congestion-avoidance algorithm.

Network congestion may also be detected by receiving more than two or three duplicate ACK packets, which are sent when packets are received out of order. When that happens, TCP performs a fast retransmit the missing packet without waiting for the timeout to expire and fast recovery that is follow the congestion-avoidance mechanism without resetting the congestion window back to 1.

Clearly, TCP provides a mechanism for reliable end-to-end transmission without requiring any support from intermediate nodes. This is done by making certain assumptions about the network. Specifically, TCP assumes that all packet losses, or unacknowledged packets and delays are caused by congestion and that the loss rate is small. This assumption is not valid in a wireless network, where packet errors are very frequent and caused mostly by poor channel conditions. Responding to packet errors by slowing down does not solve the problem if the errors are not caused by congestion. Instead, it serves only to unnecessarily reduce the throughput. Frequent errors will lead to frequent initiation of slow-start mechanisms, keeping TCP away from achieving steady state throughput.

2. Related Work

In [1], the authors compare several schemes designed to improve the performance of TCP in ad-hoc networks and categorizes these schemes into three broad categories: end-to-end protocols, link-layer protocols and split-connection protocols. Results indicate that a reliable link-layer protocol that uses knowledge of TCP to shield the sender from duplicate acknowledgments arising from wireless losses gives a 10%–30% higher throughput than one that operates independently of TCP. The split-connection approach, shields the sender from wireless losses. TCP SACK (selective acknowledgment) mechanism for the wireless hop gives better throughput and effective in dealing with a high packet loss rate when employed over the wireless multi hop environment.

The survey in [2], the authors analyze the performance of four routing protocols namely DSDV, TORA, AODV and DSR is done using ns2 simulator. A number of scenarios are generated with different mobility patterns and traffic loads of Constant Bit Rate (CBR) Traffic. The performance of each protocol is analyzed and explained the design choices that account for their performance. Results indicate that reactive routing protocols are more suitable for ad hoc networks and AODV performs almost as well as DSR at all mobility rates and movement speeds and accomplishes its goal of eliminating source routing overhead, but it still requires the transmission of many routing overhead packets and at high rates of node mobility is actually more expensive than DSR.

In [3], the authors proposed an available Bandwidth Estimation TCPW-BR a refinement of TCP Westwood that allowing the management of the Efficiency/Friendliness-to-New Reno tradeoff. TCP Westwood design adheres to the end-to-end transparency and requires only sender side modification. The key innovation of TCPW is to use a bandwidth estimate directly to drive cwin and ssthresh. The current estimation method in TCPW is based on “bandwidth Estimation”, i.e., BE. This TCPW BE strategy provides significant throughput gains, especially the large leaky pipes. Under certain congestion circumstances, BE exceeds the fair share of a connection resulting in possible unfriendliness to TCP New Reno connections.

In [4], a survey on MANET routing protocols has been done categorizing unicast, multicast and broadcast routing algorithms. Unicast algorithms are further categorized as reactive, proactive and hybrid routing algorithms. If two mobile nodes are within each other’s transmission range, they can communicate with each other directly; otherwise, the nodes in between have to forward the packets for them. In such a case, every mobile node has to function as a router to forward the packets for others. Thus, routing is a basic operation for the MANET. Because traditional routing protocols cannot be directly applied in the MANET, a lot of routing protocols for unicast, multicast, and broadcast transmission have been proposed since the advent of the MANET. This survey gives a thorough study of routing protocols in the MANET.

The Authors in [5] did comparative analysis of AODV and OLSR for average throughput as performance metric under varying network load and pause time in ns-2 simulator scenario. The result shows that the AODV perform better than OLSR for average throughput. Because in case of OLSR periodic traffic control increase cause overhead in massage. OLSR each node selects a set of Multipoint Relays (MRP) from its neighbors. The radio range of the MRP set such that it should cover all two hops neighbors. Each node has the knowledge as to for which node it acts as a MRP. Thus OLSR requires bidirectional links. OLSR is suitable for network where frequent communication take place in collection of nodes rather than as a whole.

In [6], the authors discussed the various routing protocols used in MANETs and also their weaknesses. Routing is a challenging issue in mobile ad-hoc network. This paper basically analyzes most of routing protocol available in literature. In this, DSDV, AODV, ZRP, DSR, TORA, FSR, OLSR and many other protocols are discussed. AODV is a combination of both DSR and DSDV. AODV provides both multicast, and unicast connectivity in a mobile ad-hoc environment.

In [7], the authors evaluates the performance of six TCP protocol namely- TCP-Tahoe, TCP-Reno, TCP-New Reno, TCP-Vegas, TCP-Sack and TCP-Fack for Dynamic Source Routing (DSR), Destination Sequence Distance Vector (DSDV), Ad-hoc On Demand Distance Vector (AODV) and also compare the performance of different routing protocols with varying speed & maximum packets size. Results indicate that TCP-New Reno, TCP-Vegas and TCP-Tahoe are better than other TCP variant in all considered scenarios of varying speeds and packet size.

In [8], the authors propose an extended version of regular TCP Westwood for multiple paths over wireless networks, called Multipath TCP Westwood (MPTCP) TCP Westwood (TCPW) uses the available bandwidth estimation technique to improve TCP performance in such environment. The performance do regular TCP is very poor in wireless networks, where packet loss often is caused by random error rather than by network congestion as in wired networks.

TCP Westwood (TCPW) uses the available bandwidth estimation technique to improve TCP performance in such environment. MPTCPW congestion control is designed as a coordinated control between paths which allows load-balancing feature between paths, fair sharing to regular TCPW at bottleneck. The authors also conclude that MPTCPW can achieve stability, higher throughput compared with MPTCP, fairness to regular TCPW, and greater load-balancing than uncoordinated MPTCPW under various network conditions.

3. Comparison between TCP Variants

TCP primary purpose is to provide a connection oriented reliable data transfer service between different applications to be able to provide these services on top of an unreliable communication system. TCP needs to consider data transfer, reliability flow control, multiplexing, TCP segment, and congestion control and connection management.

TCP does not depend on the underlying network layers and, hence, design of various TCP versions is based on the properties of wired networks. However, TCP congestion control algorithms may not perform well in heterogeneous networks [7]. The TCP protocol has been extensively tuned to give good performance at the transport layer in the traditional wired network environment. However, TCP in its present form is not well suited for mobile ad hoc networks (MANETs) where packet loss due to broken routes can result in the counterproductive invocation of TCP‘s congestion control mechanisms.

3.1 TCP Reno

TCP Reno employs the basic principle of Tahoe, such as slow starts and the congestion avoidance. 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 [7]. 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 an algorithm called Fast Re-Transmit. Whenever we receive 3 duplicate ACKs we take it as a sign that the segment was lost, so we re-transmit the segment without waiting for timeout. Thus we manage to re-transmit the segment with the pipe almost full. Another modification that RENO makes is in that after a packet loss, it does not reduce the congestion window to 1. Since this empties the pipe. It enters into an algorithm which we call Fast-Retransmit**.**

3.2. TCP New Reno

TCP New RENO is a slight modification over TCP-RENO. It is able to detect multiple packet losses and thus is much more efficient that RENO in the event of multiple packet losses. Like RENO, New-RENO [7] also enters into fast-retransmit when it receives multiple duplicate packets, however it differs from RENO in that it does not exit fast-recovery until all the data which was out standing at the time it entered fast recovery is acknowledged. The fast-recovery phase proceeds as in Reno, however when a fresh ACK is received then there are two cases:

- If it ACKs all the segments which were outstanding when we entered fast recovery then it exits fast recovery and sets CWD to threshold value and continues congestion avoidance like Tahoe.
- If the ACK is a partial ACK then it deduces that the next segment in line was lost and it retransmits that segment and sets the number of duplicate ACKS received to zero. It exits Fast recovery when all the data in the window is acknowledged.

3.3. TCP Westwood

TCP Westwood is a sender-side-only modification to new Reno that is intended to better handle large bandwidth-delay product paths , with potential packet loss due to transmission or other errors (leaky pipes), and with dynamic load. TCP Westwood relies on mining the ACK stream for information to help it better set the congestion control parameters: ssthresh and cwin [8].

In TCP Westwood, an Eligible Rate is estimated and used by the sender to update ssthresh and cwin upon loss indication, or during its Agile Probing phase, a proposed modification to the well-known Slow Start phase. In addition, a scheme called Persistent Non Congestion Detection (PNCD) has been devised to detect persistent lack of congestion and induce an Agile Probing phase to expeditiously utilize large dynamic bandwidth.

3.4 TCP Cubic

CUBIC TCP has an optimized congestion control algorithm; it comes as an improved version of BIC TCP. Presently, CUBIC is the default TCP algorithm in Linux [8].CUBIC improves scalability of TCP and assures a fair utilization of the bandwidth thanks to the enhanced window growth function. TCP CUBIC combines both additive- increase and binary search-increase techniques to achieve good scalability. CUBIC performs good performance in wired network scenarios. In addition the window- growth function of CUBIC is defined in real-time instead of RTT, so that, window-growth rate is independent of RTT. The growth function of CUBIC is determined by Cubic Parameters:

$$\boldsymbol{W(t) = C(t-K) + Wmax}$$

Here C is a CUBIC parameter; t is the elapsed time from the last window reduction. K is the time period that the above function takes to increase W to $\boldsymbol{W_{max}}$ when where is no further loss event a calculated as:

$$\sqrt[3]{\frac{\boldsymbol{W_{max}}}{\boldsymbol{C}}}$$

TCP CUBIC is mainly conducted by simulation and real tested experiments. The window-growth function of CUBIC is a CUBIC function having a similar shape to the growth function of BIC TCP [8]. CUBIC uses a cubic function for the elapsed time from the last congestion event. Cubic behaves like standard TCP when the cubic window-growth function is slower than standard TCP.

3.5. TCP Sack

TCP with Selective Acknowledgments is an extension of TCP RENO and it works around the problems face by TCP RENO 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 retransmits parts of RENO [4]. 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. SACK TCP [26] requires that segments not be acknowledged cumulatively but should be acknowledged selectively. If there are no such segments outstanding then it sends a new packet**.** Thus more than one lost segment can be sent in one RTT.

3.6. TCP Fack

FACK or Forward Acknowledgement is a special algorithm that works on top of the SACK options, and is geared at congestion controlling. FACK algorithm [5] uses information provided by SACK to add more precise control to the injection of data into the network during recovery – this is achieved by explicitly measuring the total number of bytes of data outstanding in the network.

FACK decouples congestion control from data recovery thereby attaining more precise control over the data flow in the network. The main idea of FACK algorithm is to consider the most forward selective acknowledgement sequence number as a sign that all the previous acknowledged segments were lost. This observation allows improving recovery of losses significantly.

3.7. TCP 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 [6]. 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. It uses a variation of Wang and crow crofts Tri-S scheme.
- **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 through put to the expected and when the difference goes above a certain threshold it exits slow start and enters the congestion avoidance phase.

4. Summary

In this article by reviewing the challenges to and basic concepts behind routing in MANETs and provided a thorough overview of TCP Variants in routing metrics and design considerations. Then classified many of the major contributions to the routing solutions. The protocols were selected in such a way as to highlight many different approaches to QoS routing in MANETs [9], while simultaneously covering most of the important advances in the field since the last such survey was published. After reviewed the operation, strengths, and drawbacks of these protocols and TCP variants in order to enunciate the variety of approaches proposed and to expose the trends in designers' thinking. The protocols' interactions with the MAC layer were also described. Finally, paper is provided an overview of the areas and trends of progress in the field and identified topics for future research.

5. Future Work

The TCP proposed mechanisms are assessed against TCP New RENO, TCP Westwood, TCP CUBIC and specific queue management algorithm to see how they fare against congestion and higher offered load. Ns2 simulator [10] is selected as the simulation tool because of the ease of use of the graphical interface provided and extensive support of TCP.

Different TCP variants react with different types of behavior. In addition, from the perspective of transport layer, it is observe that TCP will be on top of the routing protocols for reliable data transmission. Since TCP has its variants, namely TCP-New Reno, TCP-Westwood and TCP-CUBIC, In future this work could be extended to compare the TCP variants like TCP-New Reno, TCP-Westwood, and TCP-CUBIC with and without QoS management technique robust random early detection scheme against DSDV and OLSR routing protocols under different network density conditions.

References

[1] H.Balakrishnan and V. N. Padmanabhan *"A Comparison of Mechanisms for Improving TCP Performance over Wireless Links"* IEEE/ACM Transactions on networking, vol. 5, no. 6, December 1997.

[2] J.Broch, A.David, B.David, Y.C. Hu.Johnson and J.Jetcheva, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing protocols", ACM Mobicom Conference, pp no. 86-97, 1998.

[3] R.Wang, M.Valla, M. Y. Sanadid, and M.Gerla "Efficiency/Friendliness Tradeoffs in TCP Westwood" Proceedings of the Seventh International Symposium on Computers and Communications (ISCC'02), IEEE, 2002, pp. 304-311.

[4] H.Zhou, "A survey on routing protocols in MANETs," Technical report: MSU-CSE-03-08, March 2003.

[5] I. K. Tabash, N. Ahmad, and S. Beg, "Performance Evaluation of TCP Reno and Vegas over different routing protocols for MANETs" IEEE 4th International Symposium on Advanced Networks and Telecommunication Systems, 2010.

[6] S. Waghmare, A. Parab, P. Nikose, and S. J. Bhosale, "Comparative Analysis of different TCP variants in a wireless Environment" IEEE, November,2011.

[7] S. K. Kaushik, K. Chahal, S. Dhariwal and P. Raj "Analysis of TCP Performance over Mobile Ad-Hoc Networks with varying Speed" International Conference on Computing Sciences, pp. 193-200, March 2012.

[8] P. Goyel, "Simulation Study of Comparative Performance of AODV, OLSR, FSR & LAR, Routing Protocols in MANET in Large Scale Scenarios", World Congress on Information and Communication Technologies(WICT), pp no.283-286,2012.

[9] M.Ali,B.G.Stewart,A.Shahrabi and A.Vallavaraj " Congestion Adaptive Multipath Routing For Load Balancing In Mobile Adhoc Networks",International Conference on Innovations in Information Technology (IIT),IEEE,2012

[10] NetworkSimulator,http://www.isi.edu/nsnam/ns.