Analytical study of Improvement in MANET with AODV and OLSR by using TCP Vegas

Umang J. Modi[1]
[1]Computer Science and Engineering,
[1]Mewar University, Gangrar, Chittorgarh, Rajasthan

Abstract--- We exaggeration the performance of the TCP protocol. Provided further, work on using TCP (Transmission Control Protocol) to provide reliable data transmission has been performed for the purpose of smooth integration with the wired Internet. In the wired Internet, TCP-Vegas is a well-known transport protocol which takes into account existing network conditions. In this research paper I perform the survey of the congestion control mechanism of TCP reacts adversely to packet losses due to temporarily broken routes in wireless networks. Through simulations using ns-2, we observed that TCP-Vegas-Ad Hoc outperforms the standard TCP-Vegas protocol especially under high mobility scenarios over both reactive and proactive ad hoc routing protocols such as AODV and OLSR. Compare AODV and OLSR with different network parameters and conclude solution with simulation tools OPNET, NS-2 etc.

I. INTRODUCTION OF MANETS

Wireless cellular systems have been in use since 1980s. These systems work with the support of a centralized supporting structure such as an access point [1]. The wireless users can be connected with the wireless system by the help of these access points, when they roam from one place to the other.
Recent advancements such as Bluetooth introduced a fresh type of wireless systems which is frequently known as mobile ad-hoc networks. Mobile ad-hoc networks or "short live" networks control in the nonexistence of permanent infrastructure. Mobile ad hoc network offers quick and horizontal network deployment in conditions where it is not possible otherwise. Ad-hoc is a Latin word, which means "for this or for this only." Mobile ad hoc network is an autonomous system [2] of mobile nodes connected by wireless links; each node operates as an end system and a router for all other nodes in the network.
Wireless networks can be classified in two types [2]: infrastructure network and infrastructure less (ad hoc) networks. A mobile host interacts with a bridge in the network (called base station) within its communication radius. The mobile unit can move geographically while it is communicating. When it goes out of range of one base station, it connects with new base station and starts communicating through it. This is called handoff. In this approach the base stations are fixed. A Mobile ad hoc network is a group of wireless mobile computers (or nodes); in which nodes collaborate by forwarding packets for each other to allow them to communicate outside range of direct wireless transmission. MANET is an autonomous group of mobile users that communicate over reasonably slow wireless links. The network is decentralized [1], where all network activity; including discovering the topology and delivering messages must be executed by the nodes themselves. MANET is a kind of wireless ad-hoc network and it is a self-configuring network of mobile routers (and associated hosts) connected by wireless links the union of which forms an arbitrary topology. The routers, the participating nodes act as router, are free to move randomly and manage themselves arbitrarily; thus, the network's wireless topology may change rapidly and unpredictably. Such a network may operate in a standalone fashion, or may be connected to the larger Internet [1].

II. ANALYTICAL STUDY AND COMPARISON OF AODV AND OLSR

A. Ad Hoc On demand Distance Vector

The Ad Hoc On demand Distance Vector (AODV) protocol is a distance vector routing for mobile ad-hoc networks [2] [3]. AODV is an on-demand routing approach, i.e. there are no periodical exchanges of routing information.
The protocol consists of two phases [3]:

1) Route Discovery (Path Finding)
2) Route Maintenance (Path Management).

When a node want to communicate with another node first looks for a route in its routing table. If it finds path, the communication starts immediately, otherwise the node start a path finding process [8]. The route discovery process consists of a route-request message (RREQ) which is broadcasted. If a node has a valid route to the destination, it replies to the route request with a route-reply (RREP) message. Additionally, the replying node creates a so called reverse route entry in its routing table, which contains the address of the source node, the number of hops to the source, and the next hop's address, i.e. the address of the node from which the message was received.
The next phase of the protocol is called route maintenance. It is analyze by the source node and can be subdivided into:

1) Source node route:

Source node initiates a new route discovery process.[5]

2) Destination or an intermediate node route:

A route error message (RERR) is sent to the source node. Intermediate nodes receiving a RERR update their routing table by setting the distance of the destination to infinity. If the source node receives a RERR it will initiate a new route discovery. [5] To prevent global broadcast messages AODV introduces a local connectivity management. This is done by periodical exchanges of so called HELLO messages, which are small RREP packets containing a node's address and additional information [3].

B. Optimized Link State Routing Protocol

Optimized Link State Routing (OLSR) is a proactive MANET routing protocol. Unlike AODV and OLSR reduces the number of retransmissions by providing optimal routes in terms of number of hops. For this purpose, the protocol uses MPRs (Multipoint Relays) to efficiently provide its control messages by declaring the links of neighbors within its MPRs instead of all links[8].

Only the MPRs of a node retransmit its broadcast messages, hence no extra control traffic is generated in response to link failures. OLSR is particularly perfect for large and dense networks. The path from source to destination consists of a sequence of hops through the MPRs [7].

In OLSR, a HELLO message is broadcasted to all of its nearest containing information about its neighbors and their link status and received by the nodes which are one hop away but they are not passed on to further nodes [4]. In response of HELLO messages, each node would construct its MPR Selector table. MPRs of a given node are define in the subsequent HELLO messages transmitted by this node. OLSR is designed to work in a completely distributed manner and does not require reliable transmission of control messages. The recipient of a control message can easily identify which information is up-to-date - even if the received messages are not in order.

1) IERP (Interzone Routing Protocol)

Interzone Routing Protocol (IERP), the reactive routing component of the Zone Routing Protocol (ZRP). IERP adapts existing reactive routing protocol implementations to take advantage of the known topology of each node’s surrounding r-hop neighborhood (routing zone), provided by the Intrazone Routing Protocol (IARP). The availability of routing zone routes allows IERP to suppress route queries for local destinations. When a global route discovery is required, the routing area based border cast service can be used to efficiently guide route queries outward, rather than blindly relaying errors from neighbor to neighbor. Once a route has been found, IERP can use routing zones to automatically redirect data around failed links. Similarly, suboptimal route parts can be identified and traffic re-routed along shorter routes [7]

III. TCP VEGAS IN MANETS

TCP Vegas is a new design for TCP that was introduced by Brakmo et al [26, 27]. TCP Vegas includes a modified re-transmission strategy(compared to TCP Reno) that is based on fine-grained measurements of the round-trip time (RTT) as well as new mechanisms for congestion detection during slow-start and congestion avoidance.

TCP Reno's congestion detection and control mechanisms use the loss of segments as a signal that there is congestion in the network. TCP Reno has therefore no mechanism to detect the incipient stages of congestion before losses occur and hence cannot prevent such losses. Thus, TCP Reno is reactive, as it needs to create losses to find the available bandwidth of the connection. On the contrary, TCP Vegas's congestion detection mechanism is proactive [8], that is, it tries to sense incipient congestion by observing changes in the throughput rate. Since TCP Vegas infers the congestion window adjustment policy from such throughput measurements, it may be able to reduce the sending rate before the connection experiences losses.

A. New Retransmission Mechanism

TCP Vegas introduces three changes that affect TCP's (fast) retransmission strategy. First, TCP Vegas measures the RTT for every segment sent. The measurements are based on fine- grained clock values. Using the fine-grained RTT measurements, a timeout period for each segment is computed. When a duplicate acknowledgement (ACK) is received, TCP Vegas checks whether the timeout period has expired. If so, the segment is retransmitted1. Second, when a non-duplicate ACK that is the first or second after a fast retransmission is received, TCP Vegas again checks for the expiration of the timer and may retransmit another segment. Third, in case of multiple segment loss and more than one fast retransmission, the congestion window is reduced only for the first fast retransmission.

B. Congestion Avoidance Mechanism

TCP Vegas does not continually increase the congestion window during congestion avoidance. Instead, it tries to detect incipient congestion by comparing the measured throughput to its notion of expected throughput. The congestion window is increased only if these two values are close, that is, if there is enough network capacity so that the throughput can actually be achieved. The congestion window is reduced if the measured [4].

C. Modified Slow-start Mechanism

A similar congestion detection mechanism is applied during slow-start to decide when to change to the congestion avoidance phase.

D. Algorithms used to modify TCP Vegas are [3]:

1) Congestion detection during slow-start
2) Congestion detection during congestion avoidance
3) More aggressive fast re-transmit mechanism
4) Additional retransmissions for non-duplicate ACKs
5) Prevention of multiple reductions of the congestion window in case of multiple segment loss
6) Reduction of the congestion window by only 1/4 after a recovery (instead of halving it as in the case of TCP Reno).
7) A congestion window size of two segments at initialization and after a timeout (TCP Reno sets the size of the congestion window to one segment in these situations)
8) Burst avoidance limits the number of segments that can be sent at once (that is, back- to-back) to three segments
9) Spike suppression limits the output rate to at most twice the current rate. (This algorithm is turned off by default.)

E. Key points of TCP Vegas:

1) Modified Congestion Avoidance
2) Aggressive Retransmission (use fine grained timer)
3) With dupacks and with partial acks

4) Aggressive Congestion Window Adaptation
5) With recovery and with multiple loss
6) Modified Slow-Start

F. *Modified Congestion Avoidance for TCP Vegas*

TCP Vegas Calculates the expected throughput and actual throughput (Once per RTT):

Expected Throughput = Window Size/BaseRTT

Actual Throughput = ActualSentAmount/RTT Static

The decision is applied throughput the next RTT for each received ACK as follows:

1) Increase the Tx Rate (Expected-Actual>□) • cwnd= cwnd + 1/cwnd
2) Decrease Tx Rate (Expected-Actual<□) • cwnd = cwnd - 1/cwnd
3) Tx Rate Unchanged (□<Expected-Actual<□) • cwnd = cwnd

G. *Aggressive Retransmission for TCP Vegas*

With dup acks

When Vegas receive the first dupacks or the second dupacks, it checks the fine grained timer expiry.

If timer expirers, it retransmits immediately.

With partial acks

For the first two partial acks, Vegas checks whether fine grained timer expires.

If timer expires, it retransmits immediately.

H. *Aggressive cwnd updating for TCP Vegas*

With recovery

Reduce cwnd by one quarter instead of half when it enters into recovery.

With multiple loss-

In case of multiple segment loss from a single window, it reduces the cwnd only once.

With Initial setting

cwnd is set to 2 instead of 1.

I. *Modified Slow-Start for TCP Vegas*

1) Vegas Calculates (in every alternate RTT)
2) *Expected Throughput* = Window Size/BaseRTT
3) *Actual Throughput* = ActualSentAmount/RTT

J. *Vegas conclusion*

1) Less fluctuation
2) Less fluctuation in bottleneck queue and in send rate
3) Enhanced Throughput
4) Better Utilization of bottleneck capacity
5) Unfair treatment of old connection and Ineffectiveness of congestion avoidance

IV. APPROACHES TO IMPROVE TCP PERFORMANCE

Main two approaches use a generic explicit link failure notification (ELFN) scheme, although they differ in their specific mechanisms [9]:

1) TCP-Feedback and TCP-ELFN
2) Hop-by-Hop Rate Control

A. *TCP-Feedback and TCP-ELFN*

When a link failure occurs, the node just upstream of the link sends back an explicit link failure notification to the source of every TCP connection that passes through that link.[9]

B. *Hop-by-Hop Rate Control*

TCP throughput degradation is queue build-ups caused by flooding of route request packets during periods of high mobility.[9]

V. SCENARIO

The simulator used to record the performance parameters is NS 2. In the Architecture mode of the simulator the scenario is designed in an area of 1000000 square meters.[10].

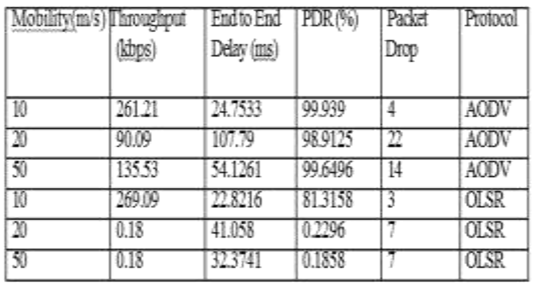
Initially if no changes are made to the area then automatically The simulator takes an area of 1000 X 1000 square meters.[10][11].

Number of Nodes density is increased from 25 to 35 in multiples. Network protocol is chosen as TCP type, we have used AODV and OLSR. [10][11].

Parameters	Value
Topology Area	1000meter*1000meter
Node Density	25, 50
Node Mobility(m/s)	10, 20, 50
Routing Protocol	AODV, OLSR
Traffic Source	TCP
Packet size	512 bytes
Simulation time	100 Sec.

Table 1: Simulator Parameters [10]

VI. TCP VEGAS - WITHOUT MODIFICATION



Mobility(m/s)	Throughput (kbps)	End to End Delay (ms)	PDR (%)	Packet Drop	Protocol
10	261.21	24.7533	99.939	4	AODV
20	90.09	107.79	98.9125	22	AODV
50	135.53	54.1261	99.6496	14	AODV
10	269.09	22.8216	81.3158	3	OLSR
20	0.18	41.058	0.2296	7	OLSR
50	0.18	32.3741	0.1858	7	OLSR

Table 2: Result for Node 25

A. *Result for Node 25*

Table 1 shows the output of performance parameters for TCP Vegas for ad hoc routing protocol AODV and OLSR; we got the result for node 25 and node mobility 10, 20 and 50 m/s respectively without any modification in TCP Vegas.

B. *Results for Node 50*

Table 2 shows the output of performance parameters for TCP Vegas for ad hoc routing protocol AODV and OLSR; we got the result for node 50 and node mobility 10, 20 and 50 m/s respectively without any modification in TCP Vegas.

Mobility(m/s)	Throughput (kbps)	End to End Delay (ms)	PDR (%)	Packet Drop	Protocol
10	123.66	44.7896	99.5959	12	AODV
20	49.22	204.716	96.2829	45	AODV
50	136.69	38.9661	99.5515	19	AODV
10	94.56	35.2648	29.2569	10	OLSR
20	0.9	46.6817	0.0393	5	OLSR
50	72.37	23.7458	22.8345	13	OLSR

Table 3: Results for Node 50

VII. TCP VEGAS - WITH MODIFICATION

A. *Modification for TCP Vegas:*

According to characteristic of TCP Vegas, it gives the minimum packet loss for MANETs but, also reduce the throughput. We made change in slow-start phase logically and vary the congestion window, because congestion window will affect the performance of TCP Vegas. We minimize the packet loss but also increase the throughput and also get changes in other performance parameters, end-to-end delay and PDR [].

B. *Result for Node 25*

Table 7.1 shows the output of performance parameters for TCP Vegas for ad hoc routing protocol AODV and OLSR; we get the result for node 25 and node mobility 10, 20 and 50 m/s respectively with some modification in TCP Vegas.
In highly competitive manufacturing industries nowadays, the manufactures ultimate goals are to produce high quality product with less cost and time constraints. To achieve these goals, one of the considerations is by optimizing the machining process parameters.

1) Optimization is the act of obtaining the best result under given circumstances. In design, construction, and maintenance of any engineering system, engineers have to take many technological and managerial decisions at several stages. The ultimate goal of all such decisions is either to minimize the effort required or to maximize the desired benefit. Since the effort required or the benefit desired in any practical situation can be expressed as a function of certain decision variables.[5]
2) Optimization can be defined as the process of finding the conditions that give the maximum or minimum value of function under given constraints. The various methods that used in optimization can be described as below.

Mobility(m/s)	Throughput (kbps)	End to End Delay (ms)	PDR (%)	Packet Drop	Protocol
10	261.66	24.8611	99.9565	1	AODV
20	91.84	67.1777	98.9125	10	AODV
50	136.16	40.2792	99.6496	15	AODV
10	269.34	20.8080	81.0607	2	OLSR
20	0.18	41.0580	0.2296	7	OLSR
50	0.18	41.058	0.2296	7	OLSR

Table 4: Results for node 25

C. *Results for Node 50*

Mobility(m/s)	Throughput (kbps)	End to End Delay (ms)	PDR (%)	Packet Drop	Protocol
10	118.96	54.227	99.2752	24	AODV
20	39.94	122.502	95.898	48	AODV
50	137.80	56.6602	99.2591	24	AODV
10	81.44	43.7703	25.7152	12	OLSR
20	0.9	46.6817	0.0393	5	OLSR
50	60.85	36.6385	20.0723	22	OLSR

Table 5: Results for node 50

D. *Analyses of Results*

We have done experiment on TCP Vegas for routing protocols AODV and OLSR using network simulator NS2, for node density 25 and 50 and node mobility 10, 20 and 50 m/s .
We got the different result for performance parameters Throughput, End to end delay, PDR, and packet drop.
We compare the two ad hoc routing protocols AODV-reactive routing protocol and OLSR- proactive routing protocol based on performance parameters, for node density 25 and 50 by taking different value of node mobility 10, 20 and 50 m/s for TCP Vegas

E. *Analysis of Results*

We have done experiment on TCP Vegas for routing protocols AODV and OLSR using network simulator NS2, for node density 25 and 50 and node mobility 10, 20 and 50 m/s .
We got the different result for performance parameters Throughput, End to end delay, PDR, and packet drop.
We compare the two ad hoc routing protocols AODV-reactive routing protocol and OLSR- proactive routing protocol based on performance parameters, for node density 25 and 50 by taking different value of node mobility 10, 20 and 50 m/s for TCP Vegas

Mobility(m/s)	Throughput	End to End Delay	PDR	Packet Drop	Protocol
10	0.17% increase	0.44% increase	0.02% increase	75% decrease	AODV
20	1.94% increase	37.68% decrease	0.86% increase	54.55% decrease	AODV
50	0.46% increase	25.58% decrease	0.14% increase	7.14% increase	AODV
10	0.09% increase	8.82% decrease	0.31% decrease	33.33% decrease	OLSR
20	NC	NC	NC	NC	OLSR
50	NC	26.82% increase	23.585 increase	NC	OLSR

Table 5: percentage change in performance parameters after modification for node 25

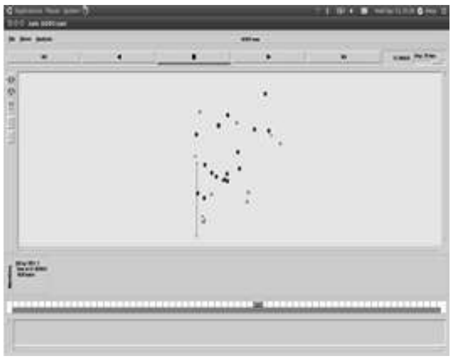


Fig. 1: Snapshot of animation for AODV

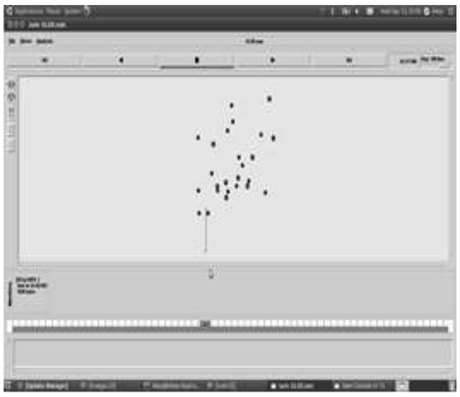


Fig. 2: Snapshot of animation for OLSR

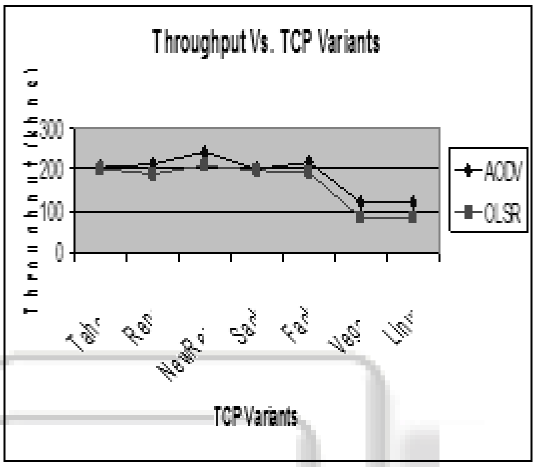


Fig. 3: Comparison Result of AODV and OLSR with TCP Vegas

VIII. GRAPHICAL ANALYSIS

A. Throughput V/s TCP Vegas

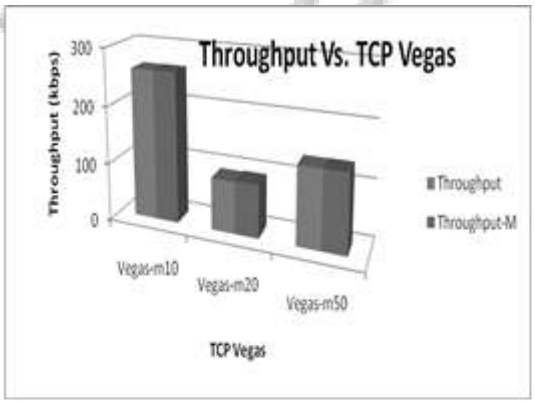


Fig. 4: Throughput V/s TCP Vegas

In routing protocols AODV, the performance parameter: throughput exhibits better results due to its path finding techniques, which uses the least and secure path in its network as compared to the other routing protocols OLSR.

B. End to End delay V/s TCP Vegas

End to end delay refers to the time taken for a packet to be transmitted across a network from source to destination Usually a data packet may take few extra second to reach the client or the server's end, which happens due to congestion in the communication network in the situation of a queue or when different routing paths are chosen by the routing protocol [6]. The graph shows the end to end delay is greatest in OLSR as compared to the others.

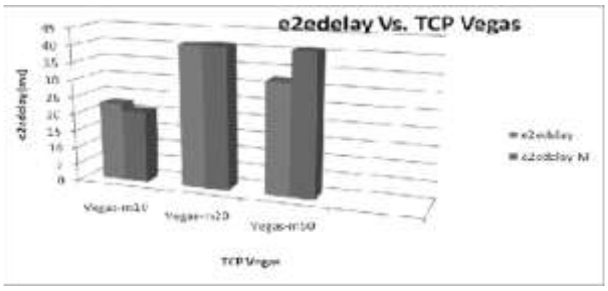


Fig. 5: End to End Delay V/s TCP Vegas

C. PDR V/s TCP Vegas

Packet delivery ratio is the segments of packets sent by source that are received by the receiver and is calculated by dividing the number of packets received by the destination through the number of packets originated by the application layer of the source [10]. Its higher value indicates best performance of the protocol. The graph is shows the best PDR is in the case of AODV as compared to OLSR.

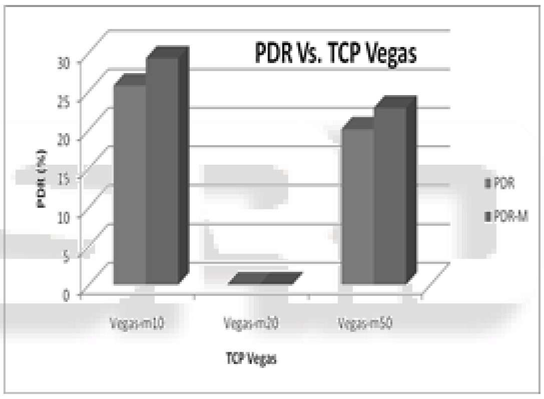


Fig. 6: PDR V/s TCP Vegas

IX. CONCLUSION

We have compared two ad hoc routing protocols, namely, Ad hoc On-Demand Distance Vector Routing (AODV) and Optimized Link State Routing (OLSR). The simulation of these protocols has been carried out using Ns-2 [7].

Three different simulation scenarios are generated; the node mobility has varied from 10 m/sec, 20 m/sec and 50 /sec. for 25 nodes and 50 nodes for TCP Vegas. Other network parameters are kept constant during the simulation.

It is observed in that above results using TCP Vegas and performance parameters like throughput, end-to-end delay, and PDR (Packet Delivery Ratio), AODV performs well in node mobility 20 m/s. but, OLSR is not suitable for node density 25 and OLSR performs well in node mobility 50 m/s. but, AODV is not suitable for node density 50.

REFERNCES

[1] Das S. Perkins C.E., Belding-Royer E.M. Ad- hoc on-demand distance vector (AODV) routing.RFC 3561, IETF Network Working Group, 2003.

[2] [Georgy Sklyarenko [MatrNr.3935701] ,AODV Routing Protocol Institutf¨urInformatik,Freie

University¨ at Berlin, Takustr. 9, D-14195 Berlin, Germany.

[3] Pucha H. Hu Y.C. Koutsonikolas D., Das S.M. On optimal ttl sequence-based route discovery in Manets. volume vol.9, p.923, 2005

[4] Srinivas Sethi , Ashima Rout , Debajyoti Mishra ,An Effective and Scalable AODV for Wireless Ad hoc Sensor Networks , Dept. of CSEA IGIT Sarang Odisha, India, Dept. of Elect. And Etc. Engg. IGIT Sarang Odisha, India. Dept. of Elect. And Etc. Engg. IGIT Sarang Odisha, India.

[5] Rajan Bansal M.Tech (CSE) BMSCE Muktsar, 152026, India, Himani Goyal Principal BMSCE Muktsar, 152026, India Parminder Singh Senior Lecturer CEC Landran, 140307, India. Analytical Study the Performance Evaluation of Mobile Ad Hoc Networks using AODV Protocol January 2011.

[6] Ramandeep Kaur, Chandan Sharma Review paper on performance analysis of AODV, DSDV, OLSR on the basis of packet delivery, IOSRJournal Volume 11, Issue 1 (May. - Jun. 2013).

[7] Ashish Allen Roberts, Rajeev Paulus, A.K. Jaiswal WSN Performance Parameters of AODV, DYMO, OLSR and IERP in RWP Mobility Model through Qualnet, IJCA Volume 65– No.22, March 2013.

[8] Thakore Mitesh C, Performance Analysis of AODV and OLSR Routing Protocol with Different Topologies, IJSR Volume 2 Issue 1, January 2013.

[9] P. Sinha, J. Monks and V. Bharghavan. Limitations of TCP-ELFN for Ad hoc networks. In Proceeding of IEEE International workshop on Mobile Multimedia Communications, October 2000.

[10] Teerawat Issariyakul, Ekram Hossain Introduction to Network Simulator NS2, TOT Public Company Limited University of Manitoba July 2008.

[11] Marc Greis' Tutorial for the UCB/LBNL/VINT Network Simulator "ns", http://www.isi.edu/nsnam/ns/tutorial/ (1 di 2) [08/01/2002 9.36.36]

[12] Vivek Kumar Me In Computer Science And Engineering Mr. Sumit Miglani (Lecterer) Simulation And Comparison Of Aodv And Dsr Routing Protocols In Manets Computer Science And Engineering Department Thapar University Patiala 147004 July 2009