Performance Analysis of TCP for Various Mobile ADHOC Routing Protocol

Mr. Ranjeet Yadav
Assistant Professor
Department of Electronics & Communication Engineering
Dr.B.R.Ambedkar Institute of Technology, Port Blair

Abstract—In this paper the performance of the TCP over different Mobile Ad-hoc Routing Protocols is evaluated by using the network simulator (NS2). The routing protocols used in the simulations are Ad-hoc On Demand Distance Vector Routing protocol (AODV), Dynamic Source Routing protocol (DSR) and Destination Sequenced Distance Vector routing protocol (DSDV). The DSDV is a table driven algorithm. DSR is source routing algorithm i.e. source appends the complete route for the packet to reach the destination in the packet's header. AODV is an on demand routing protocol. To analyze the performance of TCP over routing protocols MANET the TCP through put and TCP window size are obtained from simulations. The simulations are performed for the three routing protocols (i.e. AODV, DSDV and DSR) for the different mobility rates. By comparing the TCP throughput and TCP window size for different routing protocols at different mobility rates, the performance of the TCP for different mobile Ad-hoc Routing protocols is evaluated.

Key words: DSR - Dynamic Source Routing protocol, DSDV - Destination Sequenced Distance Vector routing protocol, AODV - Ad-hoc On Demand Distance Vector Routing protocol, ns2-Network Simulator

I. Introduction

The mobile ad-hoc network is a collection of two or more wireless nodes which might be mobile and able to communicate with each other either directly within radio range or by multi hop data forwarding operation if they are not directly within radio range. The wireless ad-hoc network is formed by any wireless devices which have networking capability and they are within radio range without any support of central administration and infrastructure. In such way, ad-hoc network has been created, organized and administered by wireless node itself on the fly. None of the wireless node has right of administration and control to support the network. Only interaction among them is used to provide such functions in a network.

According to the wireless nodes movement, ad-hoc network is classified in two major categories: Static ad-hoc network and Mobile ad-hoc network. In static ad-hoc network, location of mobile node is not frequently changed once network is deployed. In mobile ad-hoc network, all nodes are free to move without any restriction and topology of network is changing dynamically without any prior notice. This kind of network is abbreviated as MANET.

A "mobile ad hoc network" (MANET) is an autonomous system of mobile routers (and associated hosts) connected by wireless links. The routers are free to move randomly and organize themselves arbitrarily; thus, the network's wireless topology may change rapidly and unpredictably. Such a network may operate in a stand-alone fashion, or may be connected to the larger Internet. MANETs create a network among themselves dynamically without the need for any infrastructure or support from some other wired entity. Hence, we can say that Ad hoc networks are self-organizing, self-creating, self-administering and autonomous in their function. If a direction connection between one mobile node and another cannot be established, then other intermediate nodes act as routers or relays. Hence each node in a MANET acts as a host, a router, a receiver and a transmitter. In current large-scale wireless systems, this feature is absent. The enormous benefit and the potential of MANET lie in the fact that there are no costs or the need to setup an infrastructure to form such a network. Setting up traditional networks is very costly. Take the example of telephone systems where we need local loops, trunks, exchanges, which all need to be interconnected. For cellular networks, we have a number of base stations, each of which covers a small geographical area and these base stations have to communicate with a Mobile Switching Office (MSO), which acts a centralized control centre. For mobile ad hoc networks, no such costs are involved. Further, in situations like a disaster recovery site or remote areas where the fixed infrastructure based services are either not available or cannot be relied on, MANETs are the only possible solution.

II. Characteristics of Mobile Ad-hoc Networks

As ad-hoc network is formed by different devices with different capability and operates very differently, it has some unique characteristics. Some of important of ad-hoc network characteristics are explained below:

A. Dynamic Topology:

Ad-hoc network is autonomous collection of mobile nodes without any restriction. Therefore, nodes can join and depart the network without any prior notice. Also, all nodes are free to move in a network. Such property creates random, rapid and unpredictable dynamic topology in a network.

B. Limited Bandwidth:

As wireless channel has very limited bandwidth, ad-hoc network encounters very serious problem because of station role in a network. Because every station has to work as router also, station will generated more control packets in order to allow multi hop data transmission. In contrast, client of ad-hoc network demands more bandwidth.

C. Limited Power:

Most of the stations in ad-hoc networks are hand held devices which are battery operated. So, power should be consumed efficiently for longer operation of station. Also, range of transmission is also depends on power available with a node. Therefore, optimum use of power is significant design criteria for system optimization.

D. Security:

To provide security to ad-hoc network is one of the challenging tasks because of absence of central administration and fixed infrastructure. Available security techniques only reduce a threat but can not provide sufficient security in ad-hoc network. Therefore, security in ad-hoc network requires special attention.

E. Objective:

In this paper the performance of the TCP over different Mobile Ad-hoc Routing Protocols is evaluated. The routing protocol compared for the analysis are AODV, DSR and DSDV. The TCP through put and TCP window size are obtained from simulations. By comparing the TCP throughput and TCP window size for different routing protocols at different mobility rates, the performance of the TCP for different mobile Ad-hoc Routing protocols is evaluated.

III. TCP AGENTS OVERVIEW:

A. TCP Tahoe

TCP Tahoe uses three mechanisms to control the flow and deal with congestion: slow start (SS), congestion avoidance (CA), and fast retransmit. The operation of these mechanisms depends on two variables that the protocol maintains: size of the congestion window and slow start threshold (ssthresh). After the connection is initiated, the congestion window is set to 1 and the slow start threshold set to a value . The protocol enters the slow start (SS) phase in which it increases congestion window by 1 for each packet successfully acknowledged (i.e packets whose sending has been acknowledged). The effect of the SS is that congestion window grows exponentially since congestion window doubles per round-trip time (RTT). When the congestion window exceeds slow start threshold, the CA phase begins in which congestion window is increased additively by 1/cwnd for each successfully acknowledged packet.

For congestion control, TCP Tahoe uses a mechanism known as Additive Increase Multiplicative Decrease (AIMD). In this mechanism, a packet loss is taken as a sign of congestion and TCP Tahoe saves half of the congestion window as a threshold. It then sets the congestion window to one and initiates slow start until it reaches the threshold value. After that it increases linearly until it encounters a packet loss.

The problem with Tahoe is that it takes a complete timeout interval to detect a packet loss and in some cases it takes even longer due to coarse gain timeout.

B. TCP Reno

TCP Reno maintains the basic principle of Tahoe and includes mechanisms such as slow start and coarse grain re-transmit timer. But it includes an additional recovery phase. In addition to the Tahoe congestion control mechanism, Reno uses an additional algorithm known as “Fast Re-Transmit”. According to this algorithm, a packet is considered to be lost whenever three duplicate acknowledgements for that packet have been received at the sender side. In the event of a packet loss, the slow start threshold value is set to half of the current window size and the congestion window size is also set to the same value.

TCP Reno works well when the number of packet losses is small. However in the case of multiple packet losses, the protocol does not work well and its performance becomes the same as that of Tahoe under the condition of high packet loss. Furthermore, it is also possible to reduce the size of the congestion window twice for packets losses in one window.

C. TCP New Reno

TCP Reno works well in the case of single packet losses. In the case of multiple packet losses, however, it will retransmit the first packet for which it received the duplicate ACK, but then it will exit the Fast Recovery phase. After realizing that there are more dropped packets, TCP Reno will reenter the Fast Recovery phase. The constant restarting of the Fast Recovery phase affects the efficiency of the protocol. New Reno works around this problem by staying in the Fast Recovery phase as long as there are outstanding packet losses. It is implemented through the implementation of partial ACKs. Partial ACK is the acknowledgment to the first packet retransmitted in the Fast Recovery phase that has not acknowledged all the packets transmitted before the Fast Recovery phase was entered.

D. Selective Acknowledgments (SACK)

TCP SACK (selective acknowledgments) allows for finer-grain information about what packets have been lost. SACK information carried to the sender in the TCP option field of the ACK header, contains the exact information about the packets received. SACK option maintains the information about the block of packets received, i.e. the sequence numbers of the first and the last packet received within that block. If more than one packet has been dropped in one congestion window, SACK option will carry the information about several blocks differing by the sequence number of the dropped packets. This information allows TCP SACK to retransmit only the lost packets, thus providing a more aggressive loss recovery.

E. TCP Vegas

1) Algorithm:

- Define a given connection's Base RTT to be the RTT of a segment when the connection is not congested; in practice it sets Base RTT to the minimum of all measured RTTs; it is commonly the RTT of the first segment sent by the connection, before the router queues increase due to traffic generated by this connection. If we assume we are not overflowing the connection, the expected throughput is given by:

Expected = Window Size/Base RTT, where Window Size is the size of the current congestion window, which we assume to be equal to the number of bytes outstanding.

- Calculate the current Actual sending rate recording how many bytes are transmitted between the time that a segment is sent and its ACK is received and its RTT, and dividing the number of bytes transmitted by the sample RTT. This calculation is done once per round-trip time.
- Then compare Actual to Expected and adjust the window accordingly. Let Diff = Expected - Actual. Note that Diff is positive or zero by definition,

since Actual > Expected implies that we have to change Base RTT to the latest sample RTT.

- Also define two thresholds α and β, such that, α < β, roughly corresponding to having too little and too much extra data in the network, respectively.
  - When Diff < α, Vegas increases the congestion window linearly during the next RTT.
  - When Diff > β, Vegas decrease the congestion window linearly during the next RTT.
  - The congestion window is left unchanged when α < Diff < β.

Intuitively, the farther away the actual throughput gets from the expected throughput, the more congestion there is in the network, which implies that the sending rate should be reduced. The b threshold triggers this decrease. On the other hand, when the actual throughput rate gets too close to the expected throughput, the connection is in danger of not utilizing the available bandwidth. Then a threshold triggers this increase.

IV. Desirable Properties of MANET Routing Protocols

Routing protocols for mobile ad-hoc networks need to meet certain criteria in order to be considered suitable for the environment that they are functioning under. These criteria or metrics are mentioned here:

A. Distributed Operations

The protocol should be distributed in nature and not dependant on any centralized control function or node. The criterion applies to both static and mobile environments. However, in an ad hoc environment the distinguishing factor is that the nodes may enter and leave the network randomly, as well as resulting in a partitioned network due to mobility.

B. Loop Free

It is desirable that the protocol provides loop-free routes and has fail-safe mechanisms to address loop conditions. This essentially avoids any waste of precious CPU and bandwidth consumption.

C. Demand-Based Operation

The protocol must be reactive, in order to minimize the control overhead in the network, and thus conserving precious network and node resources. The protocol should only react when needed and should broadcast control information periodically.

D. Unidirectional Link Support

Most routing algorithms assume bidirectional links and do not function well under unidirectional situations. The wireless environment often causes the presence of unidirectional links, and the ability to make use of them is valuable.

E. Security

Given the nature of the wireless environment, it may be relatively simple to snoop network traffic, replay transmissions, manipulate packet headers, and redirect routing messages, within a wireless network without appropriate security provisions.

F. Power Conversation

To reduce the number of reactions to topological changes and congestion multiple routes can be used. If a particular route becomes invalid, alternate routes can be used without resorting to expensive route discovery routines.

G. Quality of Service Support

Depending on the type of application it may be required to provide Quality of Service considerations within the routing protocol. In such situation, e.g. real time traffic support, it may be necessary to incorporate the same into the routing protocol.

V. Destination Sequenced Distance Vector (DSDV) Protocol

The destination sequenced distance vector routing protocol is a proactive routing protocol. 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.

A. Protocol Overview

Each node in the network maintains routing table for the transmission of the packets and also for the connectivity to different stations in the network. These stations list for all the available destinations, and the number of hops required to reach each destination in the routing table. The routing entry is tagged with a sequence number which is originated by the destination station. In order to maintain the consistency, each station transmits and updates its routing table periodically. The packets being broadcasted between stations indicate which stations are accessible and how many hops are required to reach that particular station. Routing information is advertised by broadcasting or multicasting the packets which are transmitted periodically as when the nodes move within the network. The DSDV protocol requires that each mobile station in the network must constantly advertise to each of its neighbors, its own routing table. Since, the entries in the table my change very quickly, the advertisement should be made frequently to ensure that every node can locate its neighbors in the network. This agreement is placed, to ensure the shortest number of hops for a route to a destination; in this way the node can exchange its data even if there is no direct communication link.

The data broadcast by each node will contain its new sequence number and the following information for each new route: The destination address the number of hops required to reach the destination and the new sequence number, originally stamped by the destination. The transmitted routing tables will also contain the hardware address, network address of the mobile host transmitting them. The routing tables will contain the sequence number created by the transmitter and hence the most new destination sequence number is preferred as the basis for making forwarding decisions. This new sequence number is also updated to all the hosts in the network which may decide on how to maintain the routing entry for that originating mobile host. After receiving the route information, receiving node increments the metric and transmits information by broadcasting. Incrementing metric is done before transmission because, incoming packet will have to travel one more hop to reach its destination .Time between broadcasting the routing information packets is the other important factor to be considered. When the new information is received by the mobile host it will be retransmitted soon effecting the most rapid possible dissemination of routing information among all the cooperating mobile hosts. The mobile host cause broken links as they move form place to place within the network. The broken link may be detected by the layer2 protocol, which may be described as infinity. When the route is broken in a network, then immediately that metric is assigned an infinity metric there by determining that there is no hop and the sequence number is updated. Sequence numbers originating from the mobile hosts are defined to be even number and the sequence numbers generated to indicate infinity metrics are odd numbers. The broadcasting of the information in the DSDV protocol is of two types namely: full dump and incremental dump. Full dump broadcasting will carry all the routing information while the incremental dump will carry only information that has changed since last full dump. Irrespective of the two types, broadcasting is done in network protocol data units (NPDU). Full dump requires multiple NPDU's while incremental requires only one NPDU to fit in all the information. When an information packet is received from another node, it compares the sequence number with the available sequence number for that entry. If the sequence number is larger, then it will update the routing information with the new sequence number else if the information arrives with the same sequence number it looks for the metric entry and if the number of hops is less than the previous entry the new information is updated (if information is same or metric is more then it will discard the information). While the nodes information is being updated the metric is increased by 1 and the sequence number is also increased by 2. Similarly, if a new node enters the network, it will announce itself in the network and the nodes in the network update their routing information with a new entry for the new node. During broadcasting, the mobile hosts will transmit their routing tables periodically but due to the frequent movements by the hosts in the networks, this will lead to continuous burst of new routes transmissions upon every new sequence number from that destination. The solution for this is to delay the advertisement of such routes until it shows up a better metric.

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). Wastage of bandwidth due to unnecessary advertising of routing information even if there is no change in the network topology .DSDV doesn't support Multi path Routing. It is difficult to determine a time delay for the advertisement of routes. It is difficult to maintain the routing table's advertisement for larger network. Each and every host in the network should maintain a routing table for advertising. But for larger network this would lead to overhead, which consumes more bandwidth.

VI. AODV: AD-HOC ON-DEMAND DISTANCE VECTOR ROUTING

A. *Overview of AODV*

AODV is a reactive routing protocol. In AODV the nodes do not lie on active paths neither maintain any routing information nor participate in any periodic routing table exchanges.

Ad Hoc On-Demand Vector Routing (AODV) protocol is a reactive routing protocol for mobile ad hoc networks that maintains routes only between nodes which need to communicate. The routing messages do not contain information about the whole route path, but only about the source and the destination. Therefore, routing messages do not have an increasing size. It uses destination sequence numbers to specify how fresh a route is (in relation to another), which is used to grant loop freedom.

The Ad hoc On Demand Distance Vector (AODV) routing algorithm is a routing protocol designed for ad hoc mobile networks. AODV is capable of both unicast and multicast routing. It is an on demand algorithm, meaning that it builds routes between nodes only as desired by source nodes. It maintains these routes as long as they are needed by the sources. Additionally, AODV forms trees which connect multicast group members. The trees are composed of the group members and the nodes needed to connect the members. AODV uses sequence numbers to ensure the freshness of routes. It is loop-free, self-starting, and scales to large numbers of mobile nodes.

AODV builds routes using a route request / route reply query cycle. When a source node desires a route to a destination for which it does not already have a route, it broadcasts a route request (RREQ) packet across the network. Nodes receiving this packet update their information for the source node and set up backwards pointers to the source node in the route tables. In addition to the source node's IP address, current sequence number, and broadcast ID, the RREQ also contains the most recent sequence number for the destination of which the source node is aware. A node receiving the RREQ may send a route reply (RREP) if it is either the destination or if it has a route to the destination with corresponding sequence number greater than or equal to that contained in the RREQ. If this is the case, it unicasts a RREP back to the source. Otherwise, it rebroadcasts the RREQ. Nodes keep track of the RREQ's source IP address and broadcast ID. If they receive a RREQ which they have already processed, they discard the RREQ and do not forward it.

As the RREP propagate back to the source, nodes set up forward pointers to the destination. Once the source node receives the RREP, it may begin to forward data packets to the destination. If the source later receives a RREP containing a greater sequence number or contains the same sequence number with a smaller hop-count, it may update its routing information for that destination and begin using the better route.

As long as the route remains active, it will continue to be maintained. A route is considered active as long as there are data packets periodically traveling from the source to the destination along that path. Once the source stops sending data packets, the links will time out and eventually be deleted from the intermediate node routing tables. If a link break occurs while the route is active, the node upstream of the break propagates a route error (RERR) message to the source node to inform it of the now unreachable destination(s). After receiving the RERR, if the source node still desires the route, it can reinitiate route discovery.

VII. DSR: DYNAMIC SOURCE ROUTING

A. *Overview of DSR*

To send a packet to another host, the sender constructs a source route in the packet's header, giving the address of each host in the network through which the packet should be forwarded in order to reach the destination host. The sender then transmits the packet over its wireless network interface to the first hop identified in the source route. When a host receives a packet, if this host is not the final destination of the packet, it simply transmits the packet to the next hop identified in the source route in the packet's header. Once the packet reaches its final destination, the packet is delivered to the network layer software on that host.

Each mobile host participating in the ad hoc network maintains a route cache in which it caches source routes that it has learned. When one host sends a packet to another host, the sender first checks its route cache for a source route to the destination. If a route is found, the sender uses this route to transmit the packet. If no route is found, the sender may attempt to discover one using the route discovery protocol.

While waiting for the route discovery to complete, the host may continue normal processing and may send and receive packets with other hosts. The host may buffer the original packet in order to transmit it once the route is learned from route discovery, or it may discard the packet, relying on higher-layer protocol software to retransmit the packet if needed. Each entry in the route cache has associated with it an expiration period, after which the entry is deleted from the cache.

While a host is using any source route, it monitors the continued correct operation of that route. For example, if the sender, the destination, or any of the other hosts named as hops along a route move out of wireless transmission range of the next or previous hop along the route, the route can no longer be used to reach the destination. A route will also no longer work if any of the hosts along the route is failed or is powered off. This monitoring of the correct operation of a route in use we call route maintenance. When route maintenance detects a problem with a route in use, route discovery may be used again to discover a new, correct route to the destination.

VIII. SIMULATION EVALUATION METHODOLOGY

A. *Simulation Scenario:*

A mobile ad-hoc network with the ten nodes is considered for the simulation. The simulation is done in Network Simulator-2 (NS 2). For the simulations network area of size 500m x 500m is taken.

The initial locations of mobile nodes are as given below:
Node 0 (Source node) at (5 , 10)
Node 1 (Destination node) at (480, 270)
Node 2 at (215,235)
Node 3 at (125,110)
Node 4 at (290,290)
Node 5 at (120,10)
Node 6 at (70,230)
Node 7 at (340,15)
Node 8 at (25,120)
Node 9 at (220,120)

In these nodes node 0 is the source node and the node 1 is the destination node.

B. *Nodes mobility for the simualtions:*

For the mobile nodes mobility is considered as given below for the simulations
At time 10 seconds ,node 0 starts moving towards point (210,240)
At time 35 seconds, node 1 starts moving towards point (50,300)
At time 45 seconds, node 3 starts moving towards point (370, 20)
At time 120 seconds, node 4 starts moving towards point (390, 60)
At time 100 seconds, node 0 starts moving towards point (480, 300)

Different mobility rates of 5m/s, 10m/s, 15m/s, 20m/s, 25m/s, and 30m/s are taken in the simulations. For these different mobility rates the performance of different routing protocols is analyzed.

Parameters for the simulations are as given in the table

Parameter	**Value**
Channnel	Wireless channel
Network Interface	Phy/WirwlessPhy
MAC Type	Mac/802_11
Number of Nodes	10
Network Area	500 x 500
Interface Queue Length	50
Simulation time	160 seconds

Table 1: Simulation Parameters

C. *Performance Metrics used in the simulations:*

The TCP throughput has been used as the main metric to evaluate the performance of the ad-hoc routing protocols in these simulations. The TCP throughput for the overall network was computed according to the following:

$$\text{TCP Throughput} = \frac{\text{TCP data received at the receiver}}{\text{Duration of TCP connection}}$$

The throughput of the network in these simulations was measured for every second until the simulation was completed. Therefore the duration of TCP Connection in above equation is 1.

Another performance metric used in the simulations is the TCP window size.

D. *Simulation Results*

DSDV requires longest connection establishment time, and hence, it is slow, and does not adapt well to an ad-hoc network. Since the DSDV using the shortest path between the source and destination nodes, the TCP window size increasing fastly. During the connection the TCP Throughput is higher than the AODV. But the connection exists for short duration therefore the over all Throughput is smaller than the AODV.

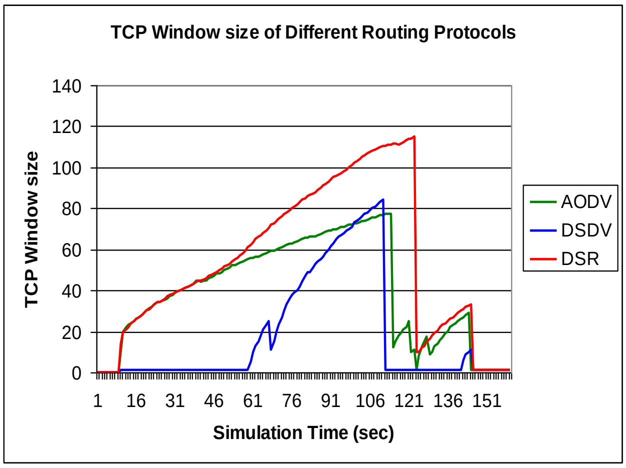

Fig. 1: TCP Window size of different Routing Protocols

- After connection is established with the AODV algorithm, the window size increases monotonously since same routing path is maintained until the available route is failed. When ever the available route is failed then the connection through the new path is established.
- DSR dynamically changes its routing path to the shortest one, and therefore it works very well in mobile ad-hoc networks. DSR achieves largest window size because of its shortest paths. DSR requires relatively short connection establishment time.

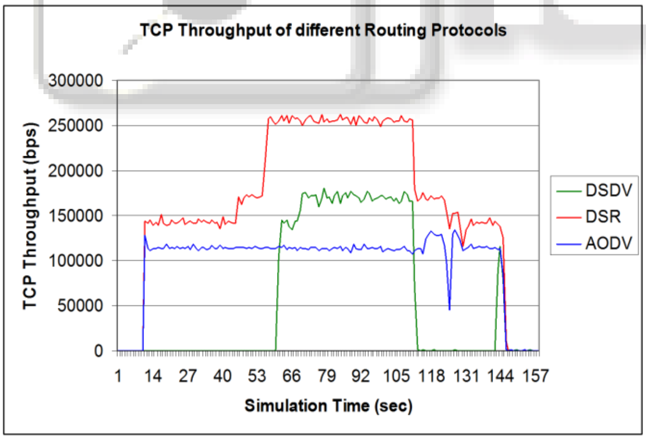

Fig. 2: TCP Throughput of different Routing Protocols

- DSDV maintains a connection only for a very short interval of time, but achieves a high throughput. But on average TCP throughput of DSDV is less than the AODV.
- AODV achieves relatively constant throughput for a longer period than DSDV.
- DSR performs well and maintains throughput levels, due to the switching of paths. DSR achieves higher throughput than AODV because the latter does not adapt the routing path sufficiently rapidly.
- In AODV the changes in paths were avoided. There were no losses due to the window remained high. However, we see that it reaches values less than the DSR. This is due to the fact that the RTT is longer since direct path is not used here.

Routing Protocol	**Connection Establishment delay**
DSDV	62 sec
AODV	11 sec
DSR	10 sec

Table 2: Connection Establishment delay for different Routing Protocols

E. *Average TCP Throughput (Kbps) Vs Node Mobility(m/s)*

Mobility rate	DSDV	AODV	DSR
Without mobility	61.847	107.19	133.73
5m/s	56.23	100.7	178.48
10m/s	54.84	97.6	163.49
15m/s	50.65	91.2	155.93
20m/s	54.22	105.5	152.12
25m/s	53.09	87.6	143.45
30m/s	52.21	86.5	151.61

Table 3: Average Throughput (Kbps) Vs Node Mobility(m/s)

From the Average Throughput analysis at different mobility rates, for different routing protocols it is observe that

- DSDV performs poor in the Mobile Ad-hoc networks with high node mobility.
- AODV Average throughput is better than DSDV.
- DSR performs well in the Mobile Ad-hoc Networks and at all mobility rates. Therefore the DSR is best for the mobile Ad-hoc networks at higher mobility rates.

IX. Conclusion

In this project the performance of the TCP over different Mobile Ad-hoc Routing Protocols is evaluated by using the network simulator (NS2). To analyze the performance of TCP over routing protocols, MANET with 10 nodes is considered for the simulations. The simulations are run for the 160 seconds. The TCP through put and TCP window size are obtained from simulations. The average throughput is also calculated at different mobility rates.

DSDV performs well when the node mobility is low. DSDV has less TCP Throughput when compared with the other two protocols. During the connection the DSDV is giving the higher Throughput than the AODV. Therefore if the node mobility is not there after this and the connection exists for a longer period the DSDV performs better than the AODV.

AODV achieves relatively constant Throughput for a longer period than DSDV. Once the connection is established the window size is increases monotonously since same routing path is maintained. The switching to the shortest path in not presented in this therefore it performs well for the moderate mobility rates.

DSR performs well at almost at all mobility rates it is giving the better performance. It is having relatively low connection establishment time. The TCP Throughput of DSR is high since it is choosing shortest path. Therefore DSR is suitable for the mobile ad-hoc networks with the higher mobility rates.

REFERENCES

[1] A.S.Tanenbaum, COMPUTER NETWORKS, 3rd Edition, PHI publications.

[2] High performance TCP/IP Networking – Mahbub Hassan, Raj Jain, PHI, 2005.

[3] An Engineering Approach to Computer Networks-S.Keshar

[4] H. Lim, K. Xu, and M.Gerla, "TCP performance over multipath routing in mobile ad hoc networks," Proceedings of IEEE ICC 2003, vol. 2, pp. 1064–1068, May 2003.

[5] S. R. Das, C. E. Perkins, and E. M. Royer, "Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks," Proceedings of the IEEE Infocom, pp. 3–12, March 2000.

[6] Performance evaluation Of TCP over routing protocols for mobile Ad-hoc Networks. Md. Abdullah-Al-Mamun, M. Mahbubur Rahman, Department of Information and Communication Engineering, Islamic University, Bangladesh

[7] Josh Broch, David B.Johnson,and David A.Maltz. The Dynamic Source Routing Protocol for Mobile Ad Hoc networks. Internet-Draft, draft-ietf-manet-dsr-00.txt,

[8] G. Holland and N. Vaidya, "Analysis of TCP performance over mobile ad hoc networks," Proceedings of ACM Mobicom, pp. 219–230,

[9] C. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," in Proceedings of the ACM SIGCOMM'94 Conference on Communications Architectures, Protocols and Applications, London, UK, pp. 234–244.

[10] "The VINT Project," USC/ISI, Xerox PARC, LBNL, and UC Berkeley. [Online]. Available: http://www.isi.edu/nsnam/vint/

[11] C.E. Perkins and S.R.Das , " Ad hoc On-Demand Distance Vector (AODV) Routing", IETF MANET Working Group Internet-Draft, available on http://www.ietf/internet-drafts/draft-ietf-manet-aodv-11.txt,2002.

[12] Charles Perkins. Ad Hoc On Demand Distance Vector (AODV) routing. Internet-Draft, draft-ietf-manet-aodv-00.txt

[13] Network Simulator -2. http://www.isi.edu/nsnam/ns/

[14] NS2 Documentation Website: http://www.isi.edu/nsnam/ns/nsdocumentation