Simulation and Comparison of Communication Protocols in Ad Hoc Network

GAURAV SETHI¹, MANOJ KUMAR ²
Department of electronics & communication
LPU, Phagwara (INDIA)¹, DAVIET, Jalandhar (INDIA)²

Abstract: - MOBILE ad hoc networks (MANETs) are collections of mobile nodes, dynamically forming a temporary network without pre-existing network infrastructure or centralized administration. In this work there is a comparison of three routing protocols using various metrics such as Delay, Jitter, Sequence number, throughput, mobility, scalability, number of packets dropped with 3-D comparison. Further metrics can be used to analysis number of packet dropped, average delays, jitters of packets, sequence number of sent packets. Furthermore recommendations are given as per results. It is concluded that DSR performs fairly well with small and medium sized network. TORA performs well with dynamic networks.

Key Words:- DSR, DSDV, TORA, NS-2, MANETS, OTcl, IMEP.

1. Introduction
MOBILE ad hoc networks are collections of mobile nodes, dynamically forming a temporary network without pre-existing network infrastructure or centralized administration. Every mobile node acts itself as a router and is responsible for discovering and maintaining routes. Further, without centralized administration, MANETs can be called autonomous. To support this kind of autonomy, the routing protocol is required to automatically adjust to frequent environment changes. Simulations are necessary in order to carry out the comparison among various MANET protocols so that we can easily figure out which protocol is best suitable under some specific and predetermined conditions.

2. Dynamic Source Routing Protocol
The Dynamic Source Routing Protocol (DSR) is a reactive routing protocol [1-3]. By the means of this protocol each node can discover dynamically a source route to any destination in the network over multiple hops.

2.1 Route discovery
If a node S wants to send to a destination node D, it needs a source route. S searches its Route Cache for a valid route to D. If the Route Cache contains a valid route, node S directly fills this route into the header of the packet and sends the data packet following this sequence of hops to the destination D. No Route Discovery is carried out in this case. If no route is found in the Route Cache, a Route Discovery is initiated. S initiates the Route Discovery by transmitting a ROUTE REQUEST message as broadcast. All nodes in transmission range will receive this message. Non-target nodes will add their address to the route record in the packet and forward the packet when received the first time. They check the request id and source node id to avoid multiple retransmissions and if their address is already included in the route record to avoid loops. The target node D sends a ROUTE REPLY when it receives a ROUTE REQUEST. In case of bidirectional links the ROUTE REPLY uses the reverse route of the ROUTE REQUEST. If the links are unidirectional, D will examine its own Route Cache for a route to S and use this to send the ROUTE REPLY to the initiator S. If no route is found in Route Cache of D, D will start its own Route Discovery, but to avoid infinite numbers of Route Discoveries it will piggyback the original ROUTE REQUEST message to its own. The initiator receiving the ROUTE REPLY adds the source route to its Route Cache. When a node starts
a Route Discovery it stocks the data packet in the send buffer. The send buffer should implement strategies to avoid buffer overflows, i.e. FIFO. The node should occasionally start a new Route Discovery for the packets in the send buffer until a route is found and the packet can be sent to the destination and removed from the buffer.

2.2 Route Maintenance
The node, which originates or forwards a packet using a source route, is responsible for confirming the receipt of the packet by the next node. S originates a packet to node D over the nodes 1 and 2. In this case S is responsible for receipt of 1, 1 for 2, and finally 2 for S. A packet is retransmitted until a receipt is received or the number of retransmissions is exceeded. This confirmation is costless for DSR by using link-level acknowledgement frame defined by IEEE 802.11 or passive acknowledgement [4]. If no confirmation is received, the node transmits a ROUTE ERROR message to the original sender indicating a broken link. The sender will remove this link from its cache and look for another source route to the destination in its cache. If the route cache contains another source route, the node sends the packet using this route. Otherwise, it will initialize a new Route Request. In order to gain performance a node may salvage a data packet that creates the ROUTE ERROR instead of discarding it. After transmitting the ROUTE ERROR the node searches its Route Cache for a new source route. It replaces the source route in the data packet with the new source route, marks the packet to be salvaged and resends the packet. Another mechanism shortens automatically a source route when intermediate hops are no longer necessary. Additional improvements are piggybacking ROUTE ERROR information on the next ROUTE REQUEST of the node and caching negative route information (broken links are cached). DSR uses the key advantage of source routing. Intermediate nodes do not need to maintain up-to-date routing information in order to route the packets they forward. There is also no need for periodic routing advertisement messages, which will lead to reduce network bandwidth overhead, particularly during periods when little or no significant host movement is taking place. Battery power is also conserved on the mobile hosts, both by not sending the advertisements and by not needing to receive them; a host could go down to sleep instead. This protocol has the advantage of learning routes by scanning for information in packets that are received. A route from A to C through B means that A learns the route to C, but also that it will learn the route to B. The source route will also mean that B learns the route to A and C and that C learns the route to A and B. This form of active learning is very good and reduces overhead in the network. However, each packet carries a slight overhead containing the source route of the packet. This overhead grows when the packet has to go through more hops to reach the destination. So the packets sent will be slightly bigger, because of the overhead. Running the interfaces in promiscuous mode is a serious security issue. Since the address filtering of the interface is turned off and all packets are scanned for information. A potential intruder could listen to all packets and scan them for useful information such as passwords and credit card numbers. Applications have to provide the security by encrypting their data packets before transmission.

3. Destination Sequenced Distance Vector Protocol
The Destination Sequenced Distance Vector Protocol (DSDV) [1-3] is a proactive, distance vector protocol, which uses the Bellmann-Ford algorithm.

3.1 Routing Table Management
The routing table in each node consists of a list of all available nodes, their metric, the next hop to destination and a sequence number generated by the destination node. The routing table is used to transmit packets through the ad hoc network. In order to keep the routing table consistent with the dynamically changing topology of an ad hoc network the nodes have to update the routing table periodically or when there is a significant change in the network. Therefore mobile nodes advertise their routing information by broadcasting a routing table update packet. The metric of an update packet starts
with metric one for one-hop neighbors and is incremented by each forwarding node and additionally the original node tags the update packet with a sequence number. The receiving nodes update their routing tables if the sequence number of the update is greater than the current one or it is equal and the metric is smaller than the current metric. Delaying the advertisement of routes until best routes have been found may minimize fluctuations of the routing table. On the other hand the spreading of the routing information has to be frequent and quick enough to guarantee the consistency of the routing tables in a dynamic network. There exist two types of update packets. One is the full dump, which contains the entire routing table and must be periodically exchanged. The other is an incremental update, which only consists of the information changed since the last full dump.

4. Temporally-Ordered Routing Algorithm – TORA
Temporally Ordered Routing Algorithm (TORA)[4] is a distributed routing protocol. The basic underlying algorithm is one in a family referred to as link reversal algorithms. TORA is designed to minimize reaction to topological changes. A key concept in its design is that control messages are typically localized to a very small set of nodes. It guarantees that all routes are loop-free (temporary loops may form), and typically provides multiple routes for any source/destination pair. It provides only the routing mechanism and depends on Internet MANET Encapsulation Protocol for other underlying functions. TORA can be separated into three basic functions: creating routes, maintaining routes, and erasing routes. The creation of routes basically assigns directions to links in an undirected network or portion of the network, building a directed acyclic graph (DAG) rooted at the destination TORA associates a height with each node in the network. All messages in the network flow downstream, from a node with higher height to a node with lower height. Routes are discovered using Query (QRY) and Update (UPD) packets. When a node with no downstream links needs a route to a destination, it will broadcast a QRY packet. This QRY packet will propagate through the network until it reaches a node that has a route or the destination itself. Such a node will then broadcast a UPD packet that contains the node height. Every node receiving this UPD packet will set its own height to a larger height than specified in the UPD message. The node will then broadcast its own UPD packet. This will result in a number of directed links from the originator of the QRY packet to the destination. This process can result in multiple routes. Maintaining routes refers to reacting to topological changes in the network in a manner such that routes to the destination are re-established within a finite time, meaning that its directed portions return to a destination-oriented graph within a finite time. Upon detection of a network partition, all links in the portion of the network that has become partitioned from the destination are marked as undirected to erase invalid routes. The erasing of routes is done using clear (CLR) messages.

5. Simulation Environment
The Network Simulator 2 (ns2)[2] is a discrete event driven simulator developed at UC Berkeley. NS-2 is used to carry out simulations.

6. Metrics
1. Delay: -: This implies the delay a packet suffers between leaving the sender application and arriving at the receiver application. In Delay we are considering Average end-to-end delay.
2. Throughput: - It is the number of received packets per TIL (Time Interval Length). In throughput we are considering following types of throughput: -
   a) Throughput of Received packets.
   b) Throughput of Dropped packets.
3. Scalability: This parameter evaluates the performance of the network as the network size increases.
4. Mobility: Mobility is the major parameter of an ad hoc network. Since an ad-hoc network is primarily characterized by its ever-changing topology, so mobility of nodes is an important consideration. Mobility of a node is a function of both speed and movement patterns.

5. Jitter: An unwanted variation of one or more signal characteristics. In jitter we are considering Jitters of Sent packets.

6. Sequence Number: Sequence number is a number given to a packet so that data arrive in sequence and not lost. In sequence we are considering Sequence Number of generated packets.

7. Cumulative number of packets dropped: This term gives us the number of total number of packets dropped.

7. Results and Discussions

7.1 DSDV Protocol

7.1.1 Scalability

Throughput of received packets

- When we increase the network size there is considerable degradation in performance. Throughput of receiving packets is very low initially, drops to further low value and then rises to nominal value. But still scalability degrades the performance of network using DSDV.

7.1.2 Mobility

Throughput of received packets

- Throughput of receiving packets is very low initially, and then rises up to 75 packets/TIL and then throughput starts increasing after 100ms to 160 packets/TIL. The Throughput of receiving packets decreases for DSDV protocol because packets are sent even before routing tables are updated and packets are dropped.

Throughput of dropping packets

- As shown in the graph below the throughput of packets dropped is very high. It is one of the worst performer when comes to packet dropped.
7.1.4 Jitter
Sent packets
- Jitter remains almost constant throughout the entire operation as far as sent packets are concerned. Initially the jitter is 0.01 sec, remains almost constant.

7.1.5 Delay
Average end to end delay
- Delay as shown is less initially. Satisfactory performance overall as far as delay is concerned. As shown in the graph as network grows delay starts increasing again because more and more entries are including in the table makes it more difficult to establish route.

7.1.6 Sequence Number
Generated packets
- As shown generated sequence number increases linearly with packet generation time. It is again loop free transmission but it is still susceptible to fix in a loop.

7.1.7 Numbers of packets dropped
- As shown in the graph below the number of packets dropped is as low as 4 packets but increases as the network grows makes it more difficult to maintain the tables’ leads to increase in the number of packets dropped. Results are more clear with 3 Dimensional figure as also shown below.
7.2 DSR Protocol

7.2.1 Scalability

- With Change in number of nodes throughput of packet received decreases but not severely affected by scalability as shown in the graph it is low initially but rises to 180 packets/TIL and then becomes constant at around 160 packets/TIL.

Fig.8: Cumulative number of packets dropped packet Vs drop event time

Fig.9: 3-Dimensional graph of number of packets dropped

7.2.2 Mobility

- Throughput of receiving packets dropped with mobility as indicated from the figure decreases and then after some time it increases to appropriate level i.e. at 160 packets/TIL.

Fig.11: throughput of receiving packets Vs simulation time

7.2.3 Throughput

Throughput of receiving packets

- In DSR, initially throughput is low because more and more control packets are being sent to discover all the routes but later throughput rises..

Fig.12: throughput of receiving packets Vs simulation time

Throughput of dropping packets

- Throughput of dropping packets i.e. no. Of packets dropped/TIL is almost zero but it rises to a value of 18 after simulation time of 150.

Fig.10: throughput of receiving packets Vs simulation time
7.2.4 Jitter
Sent packets
- Jitter is almost zero throughout the operation after initial hitch; once route is established jitter becomes almost zero. DSR performs best as far as jitter is concerned.

7.2.5 Delay
Average end to end delay
- As we know in DSR no packet is sent until full route from source to destination is established. So that is why initially delay is more but once route is established delay starts reducing as seen from graph. So the end to end delay reduces and more and more faithfully packets being send from source to destination.

7.2.6 Sequence Number
Generated packets
- DSR is again loop free as sequence number increases linearly with time but more prone to fix in a loop. But probability is negligible.

7.2.7 Number of Packets dropped
- As shown in the graph the number of packet dropped is very less in DSR as compared to all other protocols because packet only sent once route is established, which means that probability of dropping the packets in intermediate node is very less. As shown in 3-dimensional graph also numbers of packets dropped are less as compared to other protocols.
7.3 TORA Protocol

7.3.1 Scalability

Not much change in reception of packets with change in network size. As shown in the graph it rises to 60 packets/TIL and then starts decreasing after simulation time of 60sec.

An important fact obtained from the graph, which is worth noticing that throughput of receiving packets /TIL improves drastically in case of TORA.

7.3.2 Mobility

An important fact obtained from the graph, which is worth noticing that throughput of receiving packets /TIL improves drastically in case of TORA.

7.3.3 Throughput

Throughput of receiving packets
- In TORA, throughput of received packets is high during the first half of transmission but in the later phase throughput almost drops to zero.

Throughput of dropping packets
- As seen in graph packet dropped is fairly high comparable to DSDV. TORA is the worst performer as far as this metric is concerned.
7.3.4 Jitter

Sent packets

- The jitter is initially Low but rises to alarming value afterwards as indicated in the graph. This severely degrades the performance of this protocol as compared to other protocols.

7.3.5 Delay

Average end to end delay

- TORA has the worst delay characteristics because of the loss of distance information with progress. Initial delay is as high as 3 sec and it never reduces to zero.

7.3.6 Sequence Number

Generated packets

- As indicated in the graph initially sequence number increases then become constant means TORA is not entirely loop free.

7.3.7 Number of packets dropped

- The number of packets dropped is very high initially, which further increases to a very high value. TORA is worst performer among all protocols as far as number of packet dropped is concerned. As shown in 3-d graph number of packet dropped is as high as 25, which is highest among all protocols.
8. Conclusion

The major results obtained from this research work are summarized as follows:

- It is observed that delay is worst in TORA, it takes time to establish a route in DSR but once the route is established delay becomes less and number of packet dropped becomes less in case of DSR.
- TORA performs worst if we take sequence number into consideration. TORA as time increases the probability of fixing in a loop increases. Rest of protocols i.e. DSR, DSDV are loop free.
- Jitter is least in DSR shows variation in the characteristics during traveling from source to destination is least as compared to other protocols.
- With mobility throughput of received packets in TORA improves drastically. Throughput of received packets for DSR almost remains same because it always maintains alternate routes in cache from which other route can be selected in even of link change. While Throughput decreases for DSDV protocol because packets are sent even before routing tables are updated and packets are dropped.
- As far as scalability is concerned considerable degradation in performance was seen in the case of DSDV protocol. For rest of the protocols throughput of received packets remains the same.
- As far as number of packets dropped is concerned TORA performs worst. Number of packets dropped is good as far as DSR is concerned. Performance of DSDV is not good as far as this metric is concerned. DSR performs best as far as numbers of packets are concerned.

Our investigations have been focused on the comparison of three routing MANET protocols. Based upon the results following conclusions are drawn: -

- It is recommended to use TORA where there is more mobility or with dynamic networks.
- DSR is suited for medium sizes network.
- DSDV is not recommended to use with large networks and dynamic networks.
- DSDV is recommended to use with small networks as its performance degrades with scalability.

References:


