Volume 2, Issue 4, April 2012 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

MANET: Issues and Behavior Analysis of Routing Protocols

Gurpinder Singh, Asst. Prof. Jaswinder Singh

University College Of Engineering,
Punjabi University Patiala(PB.),
India.
ggurpinder_singh1989@yahoo.com

Abstract— ***Mobile Ad hoc networks(MANET) are characterized by multihop wireless connectivity, infrastructure less environment and frequently changing topology. The nodes acts as router and communicate to each other. This paper aims to provide a means of understanding the issues and protocol(OSPF,DSR,AODV,TORA.OLSR.DSDV) of MANET and investigating behavior of DSR,AODV,TORA protocol using*** **metrics Throughput and Network Load. The Behavior analysis has been done by using simulation tool opnet 14.5 which is the main simulator.**

Keywords— ***MANET, OSPF,DSR,AODV,TORA.OLSR.DSDV.***

I. Introduction

Mobile Ad-hoc Network (MANET) is a collection of wireless mobile nodes and connected in dynamic manner. Nodes forming a temporary/short-lived network without any fixed infrastructure where all nodes are free to move about arbitrarily. Nodes must behave as routers, take part in discovery and maintenance of routes to other nodes in the network.[1] Wireless links in MANET are highly error prone and can go down frequently due to mobility of nodes. Stable routing is a very critical task due to highly dynamic environment in Mobile Ad-hoc Network[2].

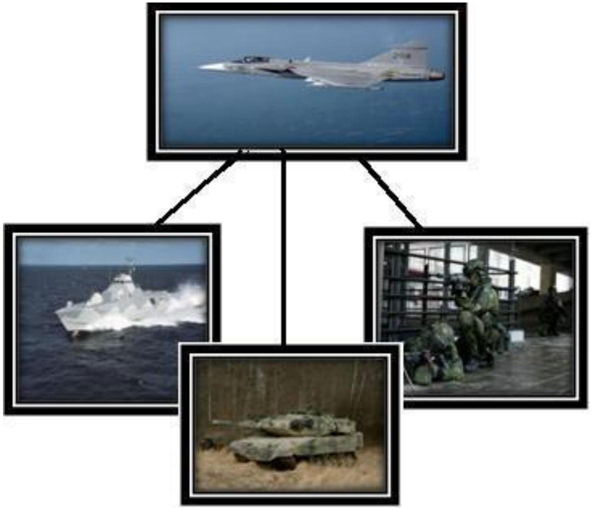


Figure1: Scenario of MANET

So mobile ad-hoc network (MANET) is a self-configuring network of mobile routers (and associated hosts) connected by wireless links - the union of which form a random topology. The routers are free to move randomly and organize themselves at random.

Such a network may operate in a standalone fashion, or may be connected to the larger Internet. Minimal configuration and quick deployment make ad hoc networks suitable for emergency situations like natural or human induced disasters, military conflicts, emergency medical situations.[3]

II. Issues in Designing MANET

Mobile Ad-hoc Network are highly dynamic in nature and no fixed infrastructure in these type of network. Due to this, issues in designing Mobile Ad-hoc Networks using a routing protocol are explain as :

A. Error-prone channel state

The characteristics of the links in a wireless network typically vary, and this calls for an interaction between the routing protocol, if necessary, find alternate routes.[4]

B. Hidden problem

Node A and node C are in range for communicating with node B, but not with each other. In the event that both try to communicate with node B simultaneously, A and C might not detect any interference on the wireless medium. Thus, the signals collide at node B, which in turn will be unable to receive the transmissions from either node. The typical solution for this so-called “Hidden terminal” problem is that the nodes coordinate transmissions themselves by asking and granting permission to send and receive packets. This scheme is often called RTS/CTS (Request To Send/Clear To Send).

The basic idea is to capture the channel by notifying other nodes about an upcoming transmission. This is done by stimulating the receiving node to output a short frame so that nearby nodes can detect that a transmission is going to take place. The nearby nodes are then expected to avoid transmitting for the duration of the upcoming (large) data frame.[4]

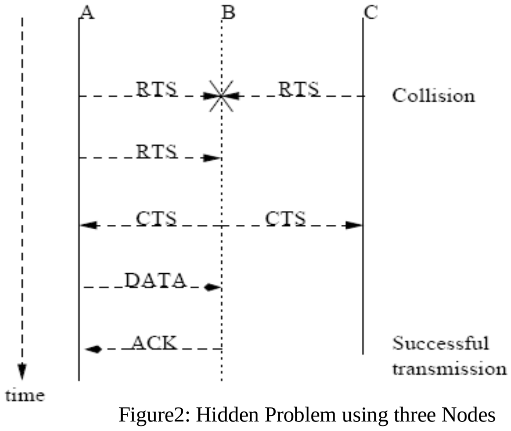


Figure2: Hidden Problem using three Nodes

C. Exposed terminals

Consider a topology similar to that of previous figure, but with an added node D only reachable from node C. Furthermore, suppose node B communicates with node A, and node C wants to transmit a packet to node D. During the transmission between node B and node A, node C senses the channel as busy. Node C falsely concludes that it may not send to node D, even though both the transmissions (i.e., between node B and node A, and between node C and node D) would succeed. Bad reception would only occur in the zone between node B and node C, where neither of the receivers is located. This problem is often referred to as “the exposed terminal problem”. Both the hidden and the exposed terminal problem cause significant reduction of network throughput when the traffic load is high.[4]

D. 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 is congestion is typically the norm rather than the exception, i.e. Aggregate application demand will likely approach or exceed network capacity frequently. As the mobile network is often simply an extension of the field network infrastructure, mobile ad hoc users will demand similar services. These demands will continue to increase as multimedia computing and collaborative networking applications rise.[5]

E. Energy-constrained operation

Some or all of the nodes in a MANET may relay on batteries or other exhaustible means for their energy. For these nodes, the most important system design criteria for optimization may be energy conservation. It should be noted that the energy consumed during sending a packet is the largest source of energy consumption of all modes. This is followed by the energy consumption during receiving a packet. Despite the fact that while in idle mode the node does not actually handle data communication operations, it has been found that the wireless interface consumes a considerable amount of energy nevertheless. This amount approaches the amount that is consumed in the receive operation. Idle energy is a wasted energy that should be eliminated or reduced through energy-efficient schemes. Through energy consumption measurements studies, experiments have also been conducted to determine the power consumption patterns in the different active modes. In some experiments, the instantaneous power consumption per communication mode, e.g. send, receive, idle and sleep modes, has been measured. Some experiments went even further to include more details about the energy consumption pattern per subtype of the operation for example, the cases of unicast and broadcast are considered to have different costs.[5]

F. Security Issues

Mobile wireless networks are generally more prone to security threats than are fixed- cable nets. The increased possibility of eavesdropping, spoofing, and denial-of-service attacks should be carefully considered. Existing link security techniques are often applied within wireless networks to reduce security threats. Snooping is unauthorized access to another person's data. It is similar to eavesdropping but is not necessarily limited to gaining access to data during its transmission. Snooping can include casual observance of an e-mail that appears on another's computer screen or watching what someone else is typing. More sophisticated snooping uses software programs to remotely monitor activity on a computer or network device. In network layer wormhole attack, a malicious node receives packets at one location in the network and tunnels them to another location in the network, where these packets are resent into the network. This tunnel between two colluding attackers is referred to as a wormhole. It could be established through wired link between two colluding attackers or through a single long-range wireless link. In this form of attack the attacker may create a wormhole even for packets not addressed to itself because of broadcast nature of the radio channel. In Black hole attack, an attacker uses the routing protocol to advertise itself as having the shortest path to the node whose packets it wants to intercept. An attacker listen the requests for routes in a flooding based protocol. When the attacker receives a request for a route to the destination node, it creates a reply consisting of an extremely short route. If the malicious reply reaches the initiating node before the reply from the actual node, a fake route gets created.[5]

III. Routing Protocols of MANET

A routing protocol is used to transmit a packet to a destination via number of nodes and numerous routing protocols have been proposed for such kind of ad- hoc networks. These protocols find a route for packet delivery and to the correct destination. Basically, routing protocols can be broadly classified into two types as:

- **Proactive protocols** In networks utilizing a proactive routing protocol, every node maintain one or more tables representing the entire topology of the network. These tables are updated regularly in order to maintain up-to-date routing information from each node to every other node. To maintain up-to-date routing information, topology information needs to be exchanged between the nodes on a regular basis which in turn leads to relatively high overhead on the network. The advantage is that routes will always be available on request.
- **Reactive protocols** Unlike proactive routing protocols, reactive routing protocols do not make the nodes initiate a route discovery process until a route is required. This leads to higher latency than with proactive protocols, but lower overhead.

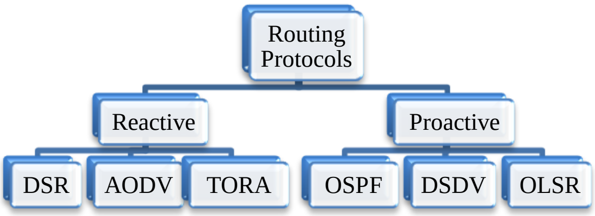


Figure3: MANET Protocols nature

A. Open Shortest Path First(OSPF)

In 2004, the OSPF Working Group at the Internet Engineering Task Force (IETF) chartered a design team to study alternatives for improving OSPF performance in such networks. During the course of the design team activities, the team has actively considered two proposals:

• A proposal known as Overlapping Relays, contributed by Madhavi Chandra (editor) from a team of authors at Cisco Systems, and

• A proposal known as MANET Designated Routers, contributed by Richard Ogier.

1)Overview of Overlapping Relays(OR): In order to support efficient flooding, routers must be aware of their 2-hop neighborhoods. ORs advertise their two-hop information in router-LSAs. A neighbor can identify its neighbor's adjacent neighbors by looking at the point-to-point links in the neighbor's router-LSA. In legacy OSPF, such a decision is not made, and a flooded LSA is reflooded by all neighbors, but with ORs each router selects a subset of its neighbor to flood its LSAs. The router tries to pick the smallest subset of neighbors that can reach all of its two-hop neighbors. Each selected router becomes an Overlapping Relay and each unselected router becomes a Non-Active Relay. When a router floods link state information, then each of its Active Relays will flood an LSA they haven't previously flooded. The OR set is merely the subset of neighbors elected to retransmit the LSAs first. An OR relays LSAs immediately upon reception, whereas a non OR (i.e., a neighbor not elected as an OR) must retransmit the LSAs if the ORs fails to do so (due to, for example, poor link quality). In situations where the OR set is not updated (the OR set was calculated before link state changes occurred) a non-OR will notice a two hop neighbor of the transmitter not receiving the LSA. It should then retransmit the LSA according the suggested algorithm for calculating the OR set can be outlined as:

- Add to the OR set the routers that will always flood LSAs.
- For all two hop neighbors reachable by only *one* one-hop neighbor, add the one-hop neighbor to the OR set.
- While there exists two-hop routers not yet covered by a node in the OR set :
    - For each one-hop neighbor, calculate the number of two-hop neighbors it can reach but that are not already covered by a router in the OR set. This number indicates the neighbor's reachability.
    - Add to the OR set, in decreasing order of willingness, one-hop neighbors that can reach at least one two-hop neighbor. If multiple neighbors have equal willingness, select the one with the highest reachability. If multiple neighbors have equal reachability, select the one with the highest number of two-hop neighbors. A final tie breaker is to select the neighbor with the highest router ID.
- Optimize the OR set by removing redundant entries, if any are found.

*2)Link Local Signaling (LLS):*To preserve the format of the existing OSPF packet format, the extra information, such information may include a list of dropped neighbors, the willingness to become OR needed to be exchanged on a link in a WOSPF-OR (Wireless Open Shortest Path First-Overlapping Relays) network is exchanged using Link Local Signaling (LLS).The existing format is preserved but the OSPF packets are appended a special data block that only routers that support LLS will consider the content of. Since all additional information needed to be exchanged in MANETs is appended to already existing OSPF packets, the routers that do not support LLS can silently ignore the LLS.

3)Overview of MANET Designated Routers: OSPF-MDRs optimize OSPF to support mobile ad hoc networks by using a Source Independent Connected Dominating Set (SI-CDS) to reduce flooding overhead and to reduce the number of adjacencies formed in dense MANETs. MDRs use Hellos to obtain 2-hop neighbor information. The Hellos contain a list of 1-way (heard) and 2-way (reported) neighbors. MDRs optimize the flooding of LSAs (Link State Advertisement)s by forming an SI-CDS. The SI-CDS consists of MANET Designated Routers (MDRs) and Backup MDRs. The

MDRs by themselves form a SI-CDS, and the MDRs and Backup MDRs together form a biconnected SI-CDS. The purpose of (Backup) MDRs in a MANET is similar to the purpose of the (Backup) DR in an OSPF broadcast network: to reduce the number of routers that must flood each LSA and to reduce the number of adjacencies. Based on 2-hop information, each router decides for itself if it will be an (B) MDR. Each MDR floods all received LSAs that have not previously been flooded. The flooding nodes do not change based on the source of the flood. BMDRs provide redundant flooding coverage. The BMDR will only flood an LSA when it did not hear an MDR flood the LSA and it did not receive acknowledges from its adjacent neighbors. MDR Other routers do not flood LSAs, but they may retransmit the LSA if it is not acknowledged. Rather than have each router form adjacencies with all of its neighbors, each (Backup) MDR forms adjacencies with a subset of its (Backup)MDR neighbors to forma biconnected backbone, and each MDR Other forms adjacencies with two selected (Backup) MDR neighbors called "parents", thus providing a biconnected sub graph of adjacencies. The parent selection is persistent, so a router updates its parents only when necessary. The persistence of the (Backup) MDRs, combined with the persistence of the parent selection, maximizes the lifetime of the adjacencies. Figure 4 depicts the number of adjacencies that would be formed in a dense network.[4]

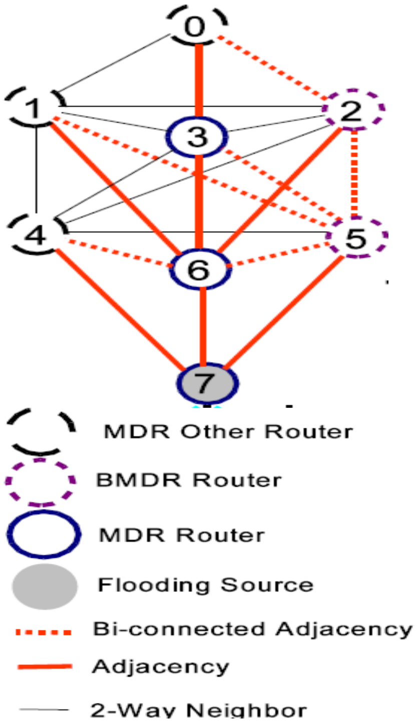


Figure 4: OSPF Network

B. Dynamic Source Routing protocol(DSR)

This is a simple and efficient routing protocol designed specifically for use in multi-hop wireless ad hoc networks of mobile nodes. Using DSR, the network is completely self-organizing and self-configuring, requiring no existing network infrastructure or administration. Network nodes (computers) cooperate to forward packets for each other to allow communication over multiple “hops” between nodes not directly within wireless transmission range of one another. As nodes in the network move about or join or leave the network, and as wireless transmission conditions such as sources of interference change, all routing is automatically determined and maintained by the DSR routing protocol. Since the number or sequence of intermediate hops needed to reach any destination may change at any time, the resulting network topology may be quite rich and rapidly changing. The DSR protocol allows nodes to dynamically discover a source route across multiple network hops to any destination in the ad hoc network. Each data packet sent then carries in its header the complete, ordered list of nodes through which the packet must pass, allowing packet routing to be trivially loop-free and avoiding the need for up-to-date routing information in the intermediate nodes through which the packet is forwarded. By including this source route in the header of each data packet, other nodes forwarding or overhearing any of these packets may also easily cache this routing information for future use. When some node ‘S’ originates a new packet destined to some other node ‘D’, it places in the header of the packet a *source route* giving the sequence of hops that the packet should follow on its way to ‘D**’**. Normally, ‘S’ will obtain a suitable source route by searching its Route Cache of routes previously learned, but if no route is found in its cache, it will initiate the Route Discovery protocol to dynamically find a new route to ‘D’. In this case, we call **S** the initiator and ‘D’ the target of the Route Discovery. For example Route Discovery, in which a node ‘A’ is attempting to discover a route to node ‘E’. To initiate the Route Discovery, ‘A’ transmits a ROUTE REQUEST message as a single local broadcast packet, which is received by (approximately) all nodes currently within wireless transmission range of ‘A’. Each ROUTE REQUEST message identifies the initiator and target of the Route Discovery, and also contains a unique request id, determined by the initiator of the REQUEST. Each ROUTE REQUEST also contains a record listing the address of each intermediate node through which this particular copy of the ROUTE REQUEST message has been forwarded. This route record is initialized to an empty list by the initiator of the Route Discovery. Route Discovery example: Node ‘A’ is the initiator, and node ‘E’ is the target. When another node receives a ROUTE REQUEST, if it is the target of the Route Discovery, it returns a ROUTE REPLY message to the initiator of the Route Discovery, giving a copy of the accumulated route record from the ROUTE REQUEST; when the initiator receives this ROUTE REPLY, it caches this route in its Route Cache for use in sending subsequent packets to this destination. Otherwise, if this node receiving the ROUTE REQUEST has recently seen another ROUTE REQUEST message from this initiator bearing this same request id, or if it finds that its own address is already listed in the route record in the ROUTE REQUEST message, it discards the REQUEST. Otherwise, this node appends its own address to the route record in the ROUTE REQUEST message and propagates it by transmitting it as a local broadcast packet (with the same request id). When initiating a Route Discovery, the sending node saves a copy of the original packet in a local buffer called the Send Buffer. The Send Buffer contains a copy of each packet that cannot be transmitted by this node because it does not yet have a source route to the packet's destination. Each packet in the Send Buffer is stamped with the time that it was placed into the Buffer and is discarded after residing in the Send Buffer for some timeout period; if necessary for preventing the Send Buffer from overflowing, a FIFO or other replacement strategy can also be used to evict packets before they expire.

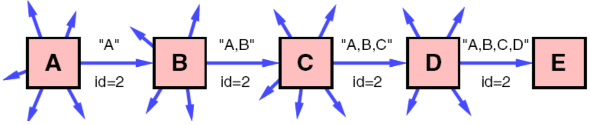
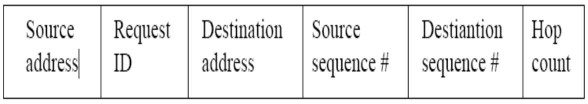


Figure 5: DSR Packet sending

The ability for nodes to reply to a ROUTE REQUEST based on information in their Route Caches, as described in, could result in a possible "ROUTE REPLY storm" in some cases. Normally, nodes would all attempt to reply from their own Route Caches, and would all send their REPLYs at about the same time since they all received the broadcast ROUTE REQUEST at about the same time. Such simultaneous replies from different nodes all receiving the ROUTE REQUEST may create packet collisions among some or all of these REPLIES and may cause local congestion in the wireless network. To solve this, nodes should delay sending its own ROUTE REPLY for a short period, while listening to see if the initiating node begins using a shorter route first. That is, this node should delay sending its own ROUTE REPLY for a random period **=H×(h-1+r)** where **h** is the length in number of network hops for the route to be returned in this node's ROUTE REPLY, **r** is a random number between 0 and 1, and **H** is a small constant delay (at least twice the link propagation delay) to be introduced per hop. This delay effectively randomizes the time at which each node sends its ROUTE REPLY, with all nodes sending ROUTE REPLYs giving routes of length less than **h** sending their REPLYs before this node, and all nodes sending ROUTE REPLYs giving routes of length greater than **h** sending their REPLYs after this node. Within the delay period, this node promiscuously receives all packets, looking for data packets from the initiator of this Route Discovery destined for the target of the Discovery.[6]

C. Ad-hoc On-demand Distance Vector (AODV)

AODV is an on-demand routing algorithm that determines a route only when a node wants to send a packet to a destination. It is a relative of the Bellman-Ford distant vector algorithm, but is adapted to work in a mobile environment. Routes are maintained as long as they are needed by the source. AODV is capable of both unicast and multicast routing. In AODV every node maintains a table containing information about which direction to send the packets in order to reach the destination.



Source address	Request ID	Destination address	Source sequence #	Destiantion sequence #	Hop count

Figure 6: AODV Route Request Packet

1) AODV Packets for path finding

- RREQ: When a route is not available for the desired destination, a route request packet is flooded throughout the network. Figure the format of such a packet.
- RREP: It a node either is, or has, a valid route to the destination, it unicast a route reply message back to the source.
- RERR: When a path breaks, the nodes on both sides of the link issue a route error to inform their end nodes of the link break.

2)Sequence Numbers in AODV: Each destination (node) maintains a monotonically increasing sequence number, which serves as a logical time at that node. Also, every route entry includes a destination sequence number, which indicates the "time" at the destination node when the route was created. The protocol uses sequence numbers to ensure that nodes only update routes with "newer" ones. Doing so, we also ensure loop-freedom for all routes to a destination. All RREQ messages include the originator's sequence number, and its (latest known) destination sequence number. Nodes receiving the RREQ add/update routes to the originator with the originator sequence number, assuming this new number is greater than that of any existing entry. If the node receives an identical RREQ message via another path, the originator sequence numbers would be the same, so in this case, the node would pick the route with the smaller hop count. If a node receiving the RREQ message has a route to the desired destination, then we use sequence numbers to determine whether this route is "fresh enough" to use as a reply to the route request. To do this, we check if this node's destination sequence number is at least as great as the maximum destination sequence number of all nodes through which the RREQ message has passed. If this is the case, then we can roughly guess that this route is not terribly out-of-date, and we send a RREP back to the originator. As with RREQ messages, RREP messages also include destination sequence numbers. This is so nodes along the route path can update their routing table entries with the latest destination sequence number.

3)Route discovery: Route discovery is initiated by issuing a RREQ message. The route is established when a RREP message is received. However, multiple RREP messages may be received, each suggesting different routes to the destination. The source only updates its path information if the RREP holds information about a more up-to-date route than already registered. Thus, every incoming RREP packet is examined to determine the most current route. When a intermediate node receives either a RREQ or a RREP packet, information about the previous node from which the packet was received is stored. This way, next time a packet following that route is received, the node knows which node is the next hop toward the source or destination, depending on which end node originated the packet.

4)Link Monitoring & Route Maintenance: Each node keeps track of a *precursor list*, and an *outgoing list*. A precursor list is a set of nodes that route through the given node. The outgoing list is the set of next-hops that this node routes through. In networks where all routes are bi-directional, these lists are essentially the same. Each node periodically sends HELLO messages to its precursors. A node decides to send a HELLO message to a given precursor only if no message has been sent to that precursor recently. Correspondingly, each node expects to periodically receive messages (not limited to HELLO messages) from each of its outgoing nodes. If a node has received no messages from some outgoing node for an extended period of time, then that node is presumed to be no longer reachable. Whenever a node determines one of its next- hops to be unreachable, it removes all affected route entries, and generates a Route Error (RERR) message. This RERR message contains a list of all destinations that have become unreachable as a result of the broken link. The node sends the RERR to each of its precursors. These precursors update their routing tables, and in turn forward the RERR to their precursors, and so on. To prevent RERR message loops, a node only forwards a RERR message if at least one route has been removed.[7]

D. Temporary Ordered Routing Protocol(TORA)

TORA is a distributed highly adaptive routing protocol designed to operate in a dynamic multihop network. TORA has four basic functions: route discovery, route maintenance, route erasing, and route optimization. TORA uses an arbitrary height parameter to determine the direction of link between any two nodes for a given destination. Consequently, multiple routes often exist for a given destination but none of them are necessarily the shortest route. To initiate a route, the node broadcasts a QUERY packet to its neighbors. This QUERY is rebroadcasted through the network until it reaches the destination or an intermediate node that has a route to the destination. The recipient of the QUERY packet then broadcasts the UPDATE packet which lists its height with respect to the destination. When this packet propagates in the network, each node that receives the UPDATE packet sets its height to a value greater than the height of the neighbor from which the UPDATE was received. This has the effect of creating a series of directed links from the original sender of the QUERY packet to the node that initially generated the UPDATE packet. When it was discovered by a node that the route to a destination is no longer valid, it will adjust its height so that it will be a local maximum with respect to its neighbors and then transmits an UPDATE packet. If the node has no neighbors of finite height with respect to the destination, then the node will attempt to discover a new route as described above. When a node detects a network partition, it will generate a CLEAR packet that results in reset of routing over the ad hoc network.[10]

E. Optimized Link State Routing protocol (OLSR)

The Optimized Link State Routing (OLSR) is a table-driven, proactive routing protocol developed for MANETs. It is an optimization of pure link state protocols that reduces the size of control packets as well as the number of control packet transmissions required. OLSR reduces the control traffic overhead by using Multipoint Relays (MPR), which is the key idea behind OLSR. An MPR is a node’s one-hop neighbor which has been chosen to forward packets. Instead of pure flooding of the network, packets are forwarded by a node’s MPRs. This delimits the network overhead, thus being more efficient than pure link state routing protocols. OLSR is well suited to large and dense mobile networks. Because of the use of MPRs, the larger and more dense a network, the more optimized link state routing is achieved. MPRs help providing the shortest path to a destination. The only requirement is that all MPRs declare the link information for their MPR selectors (i.e., the nodes which have chosen them as MPRs). The network topology information is maintained by periodically exchange link state information. If more reactivity to topological changes is required, the time interval for exchanging of link state information can be reduced. Control messages OLSR uses three kinds of control messages: HELLO, Topology Information (TC), and Multiple Interface Declaration (MID). A Hello message is sent periodically to all of a node’s neighbors. Hello messages contain information about a node’s neighbors, the nodes it has chosen as MPRs (i.e., the MPR Selector set), and a list of neighbors with whom bidirectional links have not yet been confirmed. Every node periodically floods the network with a TC message using the multipoint relaying mechanism. This message contains the node’s MPR Selector set. A MID message is used for announcing that a node is running OLSR on more than one interface. The MID message is flooded throughout the network by the MPRs. Multipoint Relays is define as, A node N selects an arbitrary subset of its 1-hop symmetric neighbors to forward data traffic. This subset, referred to as an MPR set, covers all the nodes that are two hops away. The MPR set is calculated from information about the node’s symmetric one hop and two hop neighbors. This information is extracted from HELLO messages. Similar to the MPR set, an MPR Selectors set is maintained at each node. An MPR Selector set is the set of neighbors that have chosen the node as their MPR. Upon receiving a packet, a node checks it’s MPR Selector set to see if the sender has chosen the n node as MPR. If so, the packet is forwarded, else the packet is processed and discarded. Selection of Multipoint Relay Nodes is done by choosing MPR set so that a minimum of one-hop symmetric neighbors are able to reach all the symmetric two-hop neighbors. In order to calculate the MPR set, the node must have link state information about all one-hop and two-hop neighbors. Again, this information is gathered from HELLO messages. Only nodes with willingness different than WILL_NEVER may be considered as MPR. Neighbor discovery is doing as links in an ad-hoc network can be either unidirectional or bidirectional, a protocol for determining the link status is needed. In OLSR, HELLO messages serve this purpose. HELLO messages are broadcast periodically for neighbor sensing. When a node receives a HELLO message in which its address is found, it registers the link to the source node as symmetric. As an example of how this protocol works, consider two nodes A and B which have not yet established links with each other. Firstly, A broadcasts an empty HELLO message. When B receives this message and does not find its own address, it registers in the routing table that the link to A is asymmetric. Then B broadcasts a HELLO message declaring A as an asymmetric neighbor. Upon receiving this message and finding its own address, A registers the link to B as symmetric. A then broadcasts a HELLO message declaring B as a symmetric neighbor, and B registers A as a symmetric neighbor upon reception of this message. Topology Information is Information about the network topology is extracted from *topology control* (TC) packets. These packets contain the MPR Selector set of a node, and are broadcast by every node in the network, both periodically and when changes in the MPR Selector set are detected. The packets are flooded in the network using the multipoint relaying mechanism. Every node in the network receives such TC packets, from which they extract information to build a topology table. *Route Calculation* is done by the shortest path algorithm is used for route calculations, which are initiated when a change is detected in either of the following: the link set, the neighbor set, the two-hop neighbor set, the topology set, or the Multiple Interface Association Information Base. To calculate the routing table, information is taken from the neighbor set and the topology set. The calculation is an iterative process, in which route entries are added starting from one-hop neighbors, increasing the hop count each time through. A more detailed outline in routing table is found in as:

R_dest_addr R_next_addr R_dist R_iface_addr

R_dest_addr R_next_addr R_dist R_iface_addr

Each entry in the table consists of R_dest_addr, R_next_addr, R_dist and R_iface_addr. Such entry specifies that the node identified by R_dest_addr is estimated to be R_dist hops away from the local node, that the symmetric neighbor node with interface address R_next_addr is the next hop node in the route to R_dest_addr, and that this symmetric neighbor node is reachable through the local interface with the address R_iface_addr. Entries are recorded in the routing table for each destination in the network for which a route is known. All the destinations, for which a route is broken or only partially known, are not recorded in the table.[8]-[9]

F. Destination Sequenced Distance Vector (DSDV)

It is a table-driven routing scheme for ad hoc mobile networks based on the Bellman-Ford algorithm. It was developed by C. Perkins and P. Bhagwat in 1994. The main contribution of the algorithm was to solve the routing loop problem. Each node maintains a list of all destinations and number of hops to each destination. Each entry in the routing table contains a sequence number, the sequence numbers are generally even if a link is present; else, an odd number is used. The number is generated by the destination, and the emitter needs to send out the next update with this number. Routing information is distributed between nodes by sending full dumps infrequently and smaller incremental updates more frequently. The broadcast of route updates is delayed by settling time. The only improvement made here is avoidance of routing loops in a mobile network of routers. With this improvement, routing information can always be readily available, regardless of whether the source node requires the information or not. In DSDV, a sequence number is linked to a destination node, and usually is originated by that node (the owner). The only case that a non-owner node updates a sequence number of a route is when it detects a link break on that route. An owner node always uses even-numbers as sequence numbers, and a non owner node always uses odd-numbers. With the addition of sequence numbers, routes for the same destination are selected as under:

- A route with a newer sequence number is preferred.
- In case two routes have a same sequence number, the one with a better cost metric is preferred.

The list which is maintained is called routing table with all available destinations' IP address, next hop IP address, number of hops to reach the destination, sequence number assigned by the destination node and install time. The sequence number is used to distinguish stale routes from new ones and thus avoid the formation of loops. The stations periodically transmit their routing tables to their immediate neighbors. A station also transmits its Routing table if a significant change has occurred in its table from the last update sent. So, the update is both time-driven and event-driven. As stated above one of “full dump" or an incremental update is used to send routing table updates for reducing network traffic. A full dump sends the full routing table to the neighbors and could span many packets whereas in an incremental update only those entries from the routing table are sent that has a metric change since the last update and it must fit in a packet. If there is space in the incremental update packet then those entries may be included whose sequence number has changed. When the network is relatively stable, incremental updates are sent to avoid extra traffic and full dump are relatively infrequent. In a fast-changing network, incremental packets can grow big so full dumps will be more frequent. Each route update packet, in addition to the routing table information, also contains a unique sequence number assigned by the transmitter. The route labeled with the highest (i.e. most recent) sequence number is used. If two routes have the same sequence number then the route with the best metric (i.e. shortest route) is used. Based on the past history, the stations estimate the settling time of routes. The stations delay the transmission of a routing update by settling time so as to eliminate those updates that would occur if a better route were found very soon. Each row of the update send is of the following form:

<Dest. IP Address, Dest. Sequence Number, Hop Count>[10]

IV. Simulation analysis of MANET Protocol

In this simulation we check behavior of a network using DSR, AODV, TORA protocols in scenario with 150 node, many network simulators are available to design and simulate networks in many perspectives. NS-

2 (Network Simulators-2) and OPNET (Optimized Network Engineering Tools) are the two very well-known simulators. NS-2 is open source software, OPNET is a commercial simulator but it has a comprehensive development environment to simulate network models. In the paper, simulation is performed on OPNET Modeler 14.5 [11] and the protocols DSR, AODV and TORA of MANET are used.

1) How opnet works:

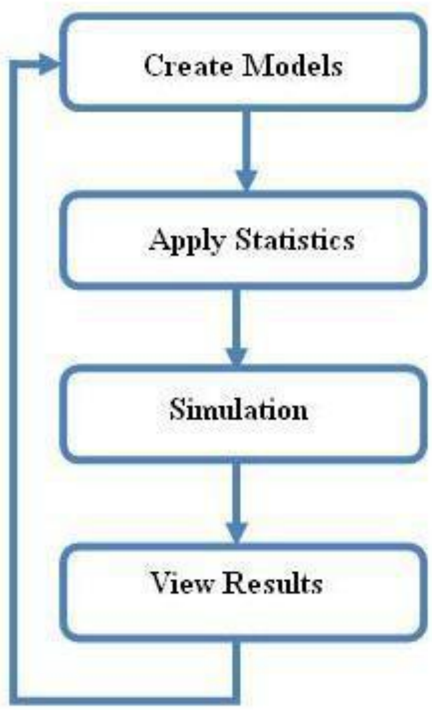


Figure 7: Working of opnet

2)Scenario values:

Routing protocol	=	DSR,AODV,OLSR
Mobility Model	=	Random Way Point Model
Terrain Size	=	1000m*1000m(Campus Area Network)
Pause Time	=	260 sec
Simulation Time	=	1800sec
Nodes	=	150

Traffic= HTTP Heavy load, HTTP Light load

3) *Simulation Results and Analysis:*

- LOAD=It is the traffic on all layer of Manet.

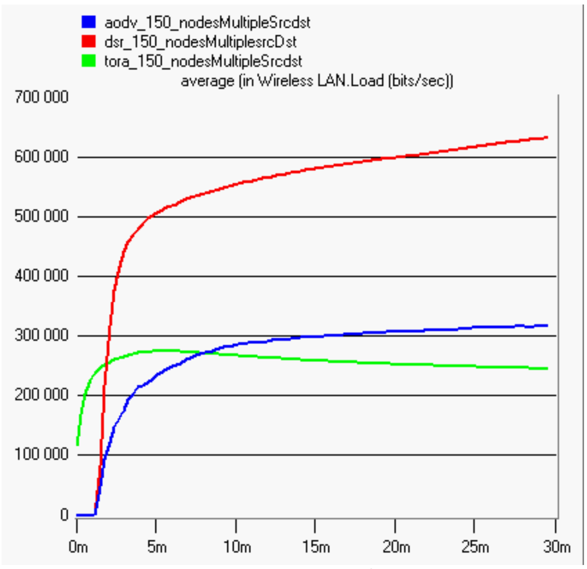


Figure 8: Wireless LAN Load for 150 nodes

As we can see that for 150 nodes TORA creates less load for network and AODV creates highest load for network. So TORA perform better for higher nodes in case of network load.

- *Throughput*= Throughput is defined as the ratio of the total data reaches a receiver from the sender. Throughput is expressed as bytes or bits per sec (byte/sec or bit/sec). A high throughput is good choice in every network.

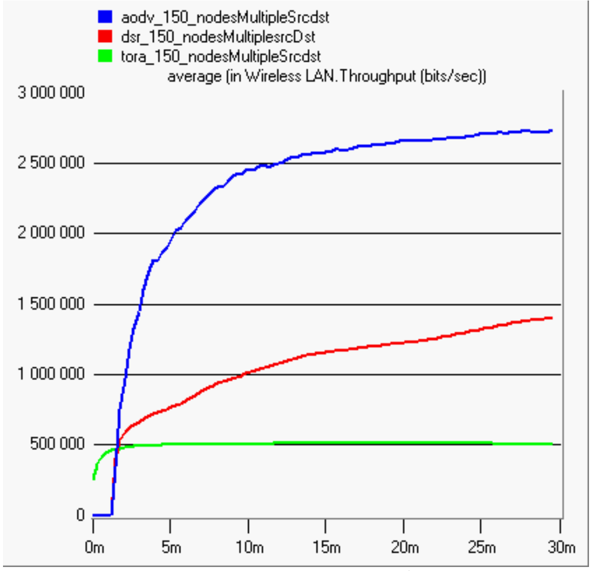


Figure 9: Wireless LAN Throughput for 150 nodes

As we can see that for 150 nodes AODV creates high throughput for network and TORA creates lowest throughput for network. So AODV perform better for higher nodes for throughput metric.

V. CONCLUSION

In this research paper, an effort has been made to concentrate on the study of routing protocols viz. OSPF,DSR,AODV,TORA,OLSR and DSDV on the basis of quantitative and qualitative metrics and also concentrate on common issues of MANET. Based on the Simulation analysis, we use DSR,AODV,TORA and we conclude that for 150 nodes TORA create less network load and throughput is high for AODV. So we conclude from that

REFERENCES

[1] Vijay Kumar1 and Ashwani Kush. "A New Scheme for Secured on Demand Routing" *IISTE Network and Complex Systems* ,Vol 2, No.2, 2012.ISSN 2224-610X (Paper), 2225-0603 (Online)

[2] Sunil Taneja & Ashwani Kush"PERFORMANCE EVALUATION OF DSR AND AODV OVER UDP AND TCP CONNECTIONS" *International Journal of Computing and Business Research (IJCBR)*, Volume 1, No. 1 December . 2010

[3] Donatas Sumyla, Mobile Ad-hoc Networks, 03/20/2006. Available: http://ecom.umfk.maine.edu/MMobile%20Ad.pdf

[4] Kenneth Holter, "Wireless Extensions to OSPF: Implementation of the Overlapping Relays Proposal",Master thesis,Department of Informatics, University of Oslo,Norway, 2nd May

[5] S. Corson & J. Macker"Mobile Ad hoc Networking:Routing Protocol Performance Issues and Evaluation Considerations", RFC 2501, Oct. 1999.
Available: http://tools.ietf.org/html/rfc2501

[6] David B. Johnson,David A. Maltz & Josh Broch"DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks" Computer Science Department Carnegie Mellon University Pittsburgh, PA 15213-3891
Available:http://www.monarch.cs.rice.edu/monarch-papers/dsr-chapter00.pdf

[7] Kenneth Holter" *Comparing AODV and OLSR*" 23rd April 2005.
Available:http://folk.uio.no/kenneho/studies/essay.pdf

[8] T. Clausen, P. Jacquet, "Optimized Link State Routing Protocol (OLSR)", RFC 3626, October 2003
Available: http://www.ietf.org/rfc/rfc3626.txt.

[9] P. Jacquet,P. Muhlethaler,T. Clausen,A. Laouiti,A. Qayyum,L. Viennot"Optimized link state routing protocol for ad-hoc networks" in International Multi Topic Conference 2001(IEEE),Dec. 2001.
Available:http://menetou.inria.fr/~muhletha/olsr.pdf

[10] Amandeep Makkar, Bharat Bhushan, Shelja, and Sunil Taneja"Behavioral Study of MANET Routing Protocols" *International Journal of Innovation, Management and Technology*, Vol. 2, No. 3, June 2011.
Available: http://www.ijimt.org/papers/133-M548.pdf

[11] OPNET Modeler 14.5,
Available: http://www.opnet.com/