CHAPTER 2

ANALYSIS OF AODV, ROUTE MAINTENANCE PARAMETERS, AND DETERMINING FACTORS

The Ad-hoc On-demand Distance Vector (AODV) routing protocol is a simple, efficient, and effective protocol for the ad-hoc network, which mainly falls under the reactive/on-demand routing. The AODV routing algorithm was actually motivated by the limited bandwidth, which is available in the media and used for communications, especially in the wireless medium. However, AODV name implies that it is an on-demand routing protocol, but it still uses characteristics of a proactive routing protocol. It takes the advantages of the DSR and DSDV algorithm; in the sense that it uses the concept of route discovery & route maintenance from DSR, and the concept of sequence numbers & sending of periodic hello messages from DSDV [R. Jain, *et al.*, 2011; & C.E. Perkins, *et al.*, 1999]. The idea of getting routes purely on-demand makes AODV routing a very useful and preferred algorithm for the ad-hoc network [C.E. Perkins *et al.*, 1996].

In this chapter, the functioning of the underlying routing protocol has been discussed in depth, i.e. the chapter explains each process that is required in the AODV network to create, delete and maintain the routes. This chapter also explains the route maintenance parameters and determining factors.

2.1 Description

AODV is a novel algorithm for the operation of ad-hoc networks. Here, each mobile node works as a specialized router, and the routes are obtained as per requirement. The algorithm of AODV routing protocol offers the self-starting, multi-hop and dynamic routing among the participating mobile nodes, which desire to deploy and maintain the ad-hoc network [C. Perkins, 1997]. AODV allows mobile nodes to adopt the routes quickly for the new destinations. And also, it allows mobile nodes to act fast in case of link breakages and changes in the network topology, in a timely manner. In the case of link breakages, AODV causes the affected set of nodes to be notified so that they are able to invalidate the routes using the lost link [K. Zahedi, *et al.*, 2011].

AODV routing uses the concept of destination sequence number in order to maintain the most recent routing information between nodes. Each node maintains a monotonically increasing sequence number counter, which is used to supersede stale cached routes. In other words, it can be said that the use of a destination sequence number guarantees that a route is fresh. Moreover, the functioning of AODV routing is also known for the prevention of loops, and by avoiding the Bellman-Ford “counting to infinity” problem, it offers quick convergence when the topology of the network changes (typically, because of node movement).

AODV uses symmetric links between neighboring nodes [C. Perkins, 1997]. It does not attempt to follow the path between nodes when one of the nodes cannot hear the other one. In this routing, due to on-demand nature, mobile nodes neither maintain routes toward the destination nor participate in any periodic routing table exchange that is not in active communication. Further, nodes do not have to discover and maintain the routes to other nodes until needed for communication or offered to serve as an intermediate forwarding node to keep connectivity between two other nodes. When the local connectivity of the mobile node is of concern, each mobile node can become aware of the other nodes in its neighborhood by the use of several techniques, including local broadcast known as “hello message”. The routing tables of the nodes within the neighborhood are organized to optimize response time to local movements and provide quick response time to a request for establishment of new routes.

In the AODV routing protocol, the change in topology causes a rapid reaction, which offers less network utilization, less processing, memory overhead and creates a unicast route to the destination within the ad-hoc network [C.K. Toh, 2001; & S.K. Sarkar, *et al.*, 2008]. Additionally, it is designed to accommodate small and large networks, even if the nodes in the network are as many as several thousands.

The key objectives of the AODV routing algorithm are as follows:

- To broadcast discovery packets only when needed.
- To differentiate between local the connectivity management, neighborhood detection and general topology maintenance.
- To spread information about changes in local connectivity to those neighboring mobile nodes that likely needs this information.

2.2 Format of Different Packets in AODV Routing

Owing to the on-demand nature of the AODV routing protocol, it seeks route when a source asks to send packets to the destination. Furthermore, the route between source and destination node once established, must be maintained for the duration of the communication. The establishment and maintenance of routes are carried out by exchanging three types of packets: Route Request (RREQ), Route Reply (RREP), and Route Error (RERR).

This section describes the format of different packets and their attributes, which are associated with it [C. Perkins, 1997].

2.2.1 Route Request (RREQ): A RREQ packet format is shown in figure (2.1), and holds the following information as:

- *J:* Join flag; reserved for the multicast.
- *R:* Repair flag; reserved for the multicast.
- *G:* Gratuitous RREP flag; a gratuitous RREP should be unicast.

- *D:* Destination only flag; only destination may respond to this RREQ.
- *U:* Unknown sequence number; destination sequence number is unknown.

<table>
<tr><td>Type-1</td><td>J</td><td>R</td><td>G</td><td>D</td><td>U</td><td>Reserved</td><td>Hop Count</td></tr>
<tr><td colspan="8">RREQ ID</td></tr>
<tr><td colspan="8">Receiver/Destination IP Address</td></tr>
<tr><td colspan="8">Receiver/Destination Sequence Number</td></tr>
<tr><td colspan="8">Sender/Source/Originator IP Address</td></tr>
<tr><td colspan="8">Sender/Source/Originator Sequence Number</td></tr>
</table>

Figure 2.1: RREQ packet format

- *Reserved:* Sent as zero; it is ignored on reception.
- *Hop Count:* It is the total number of hops from originator node to the node that is currently handling the RREQ packet.
- *Route Request ID:* It is a unique sequence number which is assigned by originating node to identify a particular RREQ packet in the network.
- *Receiver/Destination IP Address:* It is an IP address of the destination node for which a route is requested.
- *Receiver/Destination Sequence Number:* It is the latest sequence number acquired by the originator node in the past for any route towards the destination node.
- *Sender/Source/Originator IP Address:* It is an IP address of the originator node which has generated the RREQ packet.
- *Sender/Source/Originator Sequence Number:* It is the current sequence number that is used for the route entry pointing towards the originator node which has generated the RREQ packet.

2.2.2 Route Reply (RREP): A packet format of RREP is illustrated in figure (2.2), and the following information is comprised in this packet as:

- *R:* Repair flag; used for the multicast.
- *A:* Acknowledgment flag; required for acknowledgment

<table>
<tr><td>Type-2</td><td>R</td><td>A</td><td>Reserved</td><td>Prefix Size</td><td>Hop Count</td></tr>
<tr><td colspan="6">Receiver/Destination IP Address</td></tr>
<tr><td colspan="6">Receiver/Destination Sequence Number</td></tr>
<tr><td colspan="6">Sender/Source/Originator IP Address</td></tr>
<tr><td colspan="6">Lifetime</td></tr>
</table>

Figure 2.2: RREP packet format

- *Reserved:* Sent as zero; it is ignored on reception.
- *Prefix Size (5-bit):* If non-zero, it specifies that the indicated next hop may be used for any nodes with the same routing prefix as the requested destination.
- *Hop Count:* It is the total number of hops required by the packet to reach the destination node from originator node.
- *Receiver/Destination IP Address:* It is an IP address of the destination node for which a route is being provided.
- *Receiver/Destination Sequence Number:* It is the destination sequence number which is concerned with the route.
- *Sender/Source/Originator IP Address:* It is an IP address of the node which originates the RREQ packet for which the route is provided.
- *Life Time:* The time in milliseconds after which RREP packets are considered invalid.

2.2.3 Route Error (RERR): Figure (2.3) shows the RERR packet format, and this packet retains the following information as:

- *N:* No delete flag; a local repair of a link has been performed and upstream nodes should not delete the route.
- *Reserved:* Sent as zero; it is ignored on reception.
- *Destination Count:* It indicates the total number of unreachable destinations.

Type-3	N	Reserved	Destination Count
Unreachable Destination IP Address (1)			
Unreachable Destination Sequence Number (1)			
Additional Unreachable Destination IP Address (if needed)			
Additional Unreachable Destination Sequence Number (if needed)			

Figure 2.3: RERR packet format

- *Unreachable Receiver/Destination IP Address:* An IP address of the destination that has become unreachable due to a link breakage.
- *Unreachable Receiver/Destination Sequence Number:* The sequence number in the route table entry for the destination node listed in the previous unreachable destination IP address field.

2.3 AODV Routing Operation

As AODV routing protocol falls under a reactive type of routing. Therefore, its main features for routing operation are the route discovery and route maintenance process.

2.3.1 Pictorial Presentation of AODV Routing Procedure

Figure (2.4) reflects the pictorial presentation of the various processes, which is involved in AODV routing operation. From the figure, it can be seen that the RREQ, RREP, and RERR packets are mainly responsible for the AODV routing operation [C.

Perkins, 1997]. Firstly, the route discovery process is initiated by source node that floods the RREQ packet to its neighbor nodes in order to search for the best route towards the destination. In this route discovery process, the source node records the route information after it receives the RREP packet from its neighboring nodes. If the source node receives the multiple RREP packets, then the route with the minimum amount of hops is chosen. As the packets flow from source node to the destination node, the intermediate nodes update their timer that is associated with the maintenance of routes. In AODV routing, the routing table holds various information like destination/receiver address, address of the next hop, total number of hops for the route towards destination, destination sequence number, number of active neighbors for this route and the expiry time of route table entry [S.K. Sarkar, *et al.*, 2008]. Moreover, the active node in AODV routing broadcasts the “hello messages” to detect the links for any neighboring nodes. These hello messages are also used to detect link break in the network that occurs when the node fails to receive hello messages from a particular neighbor node. Whenever link failure takes place in the network, the intermediate nodes try to maintain the route at their own local level through other neighboring nodes. If it is unable to maintain the route, then it generates a RERR packet, which intimates the source with an invalid route.

Moreover, in the further coming sub-sections, the route discovery process, route maintenance process, management of routing tables, management of local connectivity, advantages, and disadvantages of the AODV routing protocol have been talked about in details.

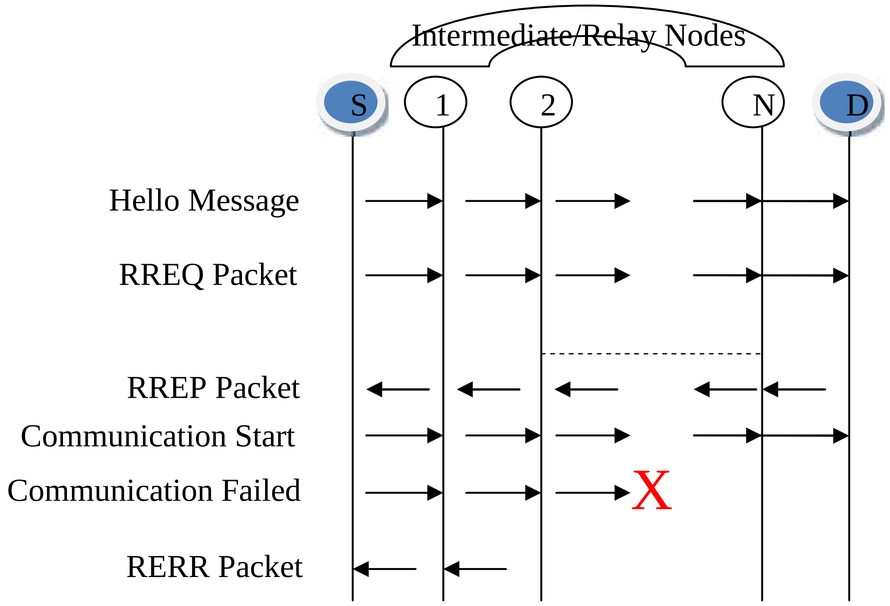

Figure 2.4: Pictorial presentation of AODV routing procedure

2.3.2 Route Discovery Process

The route discovery process is generated when a source node wishes to communicate with a destination node for which it has no information in its routing table [C. Perkins, 1997 & 2001; & C.K. Toh, 2001]. Each node maintains two separate counters: a node sequence number (a number generated by the node itself to guarantee information is up-to-date) and a broadcast_id (the unique identifier of a packet sent out by the source). The source node starts the route discovery process by sending out a RREQ packet to its neighbors.

The coupling of source address & broadcast_id gives the unique identity for a RREQ. Broadcast_id is raised each time whenever the source generates a new RREQ. Each of the neighboring nodes respond to the RREQ either by sending a RREP to the source (if it is the destination node) or by diffusing the RREQ packet to its own neighbor nodes (if it has a valid path to the requested destination) after raising the hop_count (a counter that tracks the number of hops).

It should be noted that a node could receive multiple copies of the same packet from different neighbors. When an intermediate node receives a RREQ with the same broadcast_id and source address more than once, it rejects the superfluous RREQs without re-diffusing them. Thus overloading can be avoided in the network. If a node cannot satisfy the RREQ, it keeps track of the following information to implement the reverse-path as well as the forward-path setup that will accompany the transmission of the eventual RREP [S.K. Sarkar, *et al.*, 2008].

- Destination IP address
- Source IP address
- Broadcast ID
- Expiration time for reverse-path route entry
- The source node’s sequence number.

To further limit the load on the network, AODV uses a progressive method to extend the route search. The request is firstly diffuse within a fixed number of hops. If the source does not receive a response within a set time period, another message is sent out over a wider area (i.e. the maximum number of hops is increased). If there is still no reply, this process is repeated up to a fixed maximum number of tries, after which the destination is declared unreachable.

2.3.2.1 Reverse Route Setup

As the RREQ moves from a source to different destinations, it automatically saves the reverse route from all nodes back to the source as shown in figure (2.5). This reverse route will be needed if the node receives a RREP back to the node that originates the RREQ. These reverse route entries are maintained long enough for the RREQ to cross the network and for the transmitter to receive a response. Before broadcasting the RREQ, the originating node buffers the RREQ_ID and the originating IP address. In this way, when the node receives the packet again from its neighbors, it will not reprocess and re-forward the packet.

Two sequence numbers are included in the RREQ: the sequence number of the source and the last destination sequence number known to the source. The source sequence number is used to maintain the freshness of information on the route back to the source. The destination sequence number indicates how fresh the path to the destination must be before the source accepts it.

As shown in the figure (2.5), when the source node S determines that it needs a route to the destination node D and does not have the route available, then immediately, node S starts broadcasting a RREQ packet to its neighboring nodes in quest of the route to the destination. Nodes 1 and 4 being as neighboring nodes to the node S receive the RREQ packet. Therefore, nodes 1 and 4 create a reverse link to the source from which they received RREQ. Since the nodes 1 and 4 are not aware of the link to the node D, they simply rebroadcast this RREQ to their neighboring nodes 2 and 5.

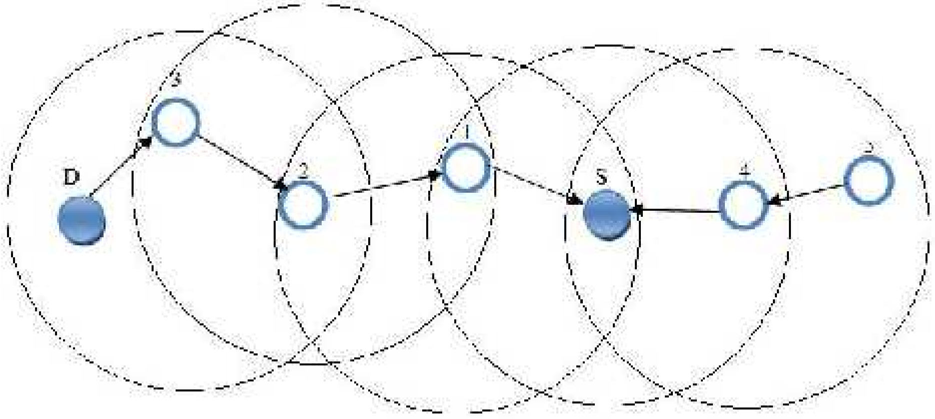

Figure 2.5: Reverse route setting

2.3.2.2 Forward Route Setup

Finally, the RREQ will arrive at a node that possesses a current route to the destination or is the destination itself. The receiving node first checks that the RREQ was received over a bi-directional link. If an intermediate node has a route entry for the required destination, it checks whether the route is current by comparing its sequence number to that of the destination in the RREQ message. If the sequence number contained in the RREQ for the destination is higher than that of the saved sequence number by the intermediary node, then the node will not use its saved route to reply the RREQ message and also, updates its sequence number for the destination. After that, the intermediary node re-diffuses the RREQ. In other words, the intermediary node may only respond if, in its routing table, it has a route for which the sequence number is greater than or equal to that contained in the RREQ. If the node has a current route to the destination and if the RREQ has not already been received, the node sends a RREP message by unicast to the neighbor node from which it received the RREQ.

When a diffusion packet reaches a node that supplies route to the destination, the return path back to the source of the RREQ has already been established. As the RREP is transmitted to the source, each node on the route saves a pointer in the exact direction of the node from which the RREP arrived, updates its routing table entries for the source & the destination and saves the latest sequence number for the requested destination.

Figure (2.6) represents the forward route setup as the RREP travels through the nodes 3, 2, 1 from the destination D to the source node S. Nodes 4 and 5 are not along the path determined by the RREP, and will timeout after ART and will delete the reverse pointers from these nodes.

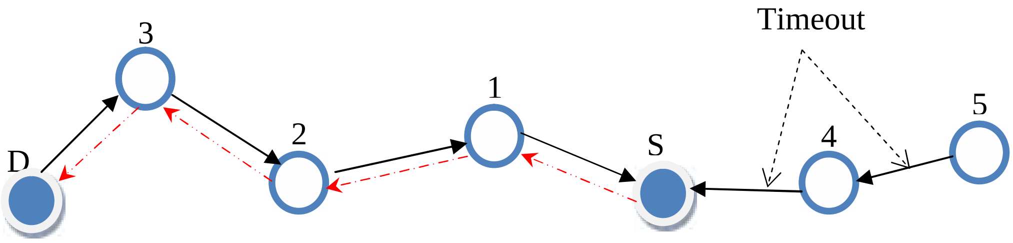

Figure 2.6: Forward route setting

The node receiving a RREP for a given source node propagates the RREP towards this source. If it receives another RREP, it updates its routing information and propagates this second RREP exclusively, if the sequence number for the destination is higher than that contained in the first RREP or if the sequence number is the same but with the lower hop_count. And, all other RREP messages will be deleted. It reduces the number of RREP messages being propagated towards the source while keeping the most up-to-date routing information and making routing as rapid as possible. Now, the source node S can begin transmitting data as soon as the first RREP is received and can update its routing information later if a better route is discovered subsequently.

2.3.2.3 Step-by-Step Procedure for Route Discovery in AODV

In this section, the step-by-step procedure for route discovery process in AODV routing has been presented as follows:

Step 1: Source node needs a route to the destination node

Step 2: Source node broadcasts the RREQ packet

Step 3: Neighbor nodes receive the RREQ packet

Step 4: Neighbor node takes the following actions after receiving the RREQ packet

```
{
    It checks whether it is a target node, i.e. destination node or not
   - If YES, then
     {
        It sends RREP packet to sender
        (Copies the accumulated route record from RREQ into RREP &
         then communication will start)
     }
- If NO, it means, it is an intermediate node
  {
  The intermediate node discards the RREQ packet
     If,
      {
          The packet has the same ID, i.e. it has already seen before
                                  Or
          Finds its own address in the route records
  (It sends RREP packet back towards the source, i.e. it has a route for
  the intending node & then communication will start)
   }
    Otherwise,
  Propagate the RREQ packet to the next neighbor nodes
  }
}
```

Step 5: Next neighbors repeat step 4

Step 6: Step 4 & 5 repeat until the intending node is found. Once the route is found towards targeting node, then communication will start.

2.3.3 Route Maintenance Process

Once a route has been established, the source node should maintain it as long as the route is needed [C. Perkins, 1997; & S.K. Sarkar, *et al.*, 2008]. The movement of nodes affects only the routes passing through this specific node and thus do not have global effects. If the source node moves while having an active session and loses connectivity with the next hop of the route, it can rebroadcast a RREQ. If though an intermediate node loses connectivity with its next hop, it initiates a RERR message, broadcasts it to its precursor nodes, and marks the entry of the destination in the route table as invalid, by setting the distance to infinity. The entry will only be discarded after a certain amount of time since routing information may still be used. When a neighbor receives the RERR message, it also marks its route table entry for the destination as invalid and sends again RERR messages to its precursor. This process continues until all active source nodes have been informed. The process will end at a fixed moment as AODV will only maintain routes without loops, and an ad-hoc network contains a finite number of nodes. At the end of the process, the source nodes may restart a route discovery process if they still require an open route to the destination concerned.

Figure (2.7.I) illustrates the initialization of RERR message. Suppose, a node N4 moves to N4' and so node N3 cannot communicate with it anymore, i.e. connectivity is lost. N3 creates a RERR message to N2, there the route is marked invalid and unicast the message to N1. The message is unicast to only those nodes through which route is passing. N1 does the same thing and unicast the message to the source node. When the RERR is received at the source node and it still needs the route to the destination, it re-initiates a route discovery process. Figure (2.7.II) shows the new route from the source to the destination through node N5.

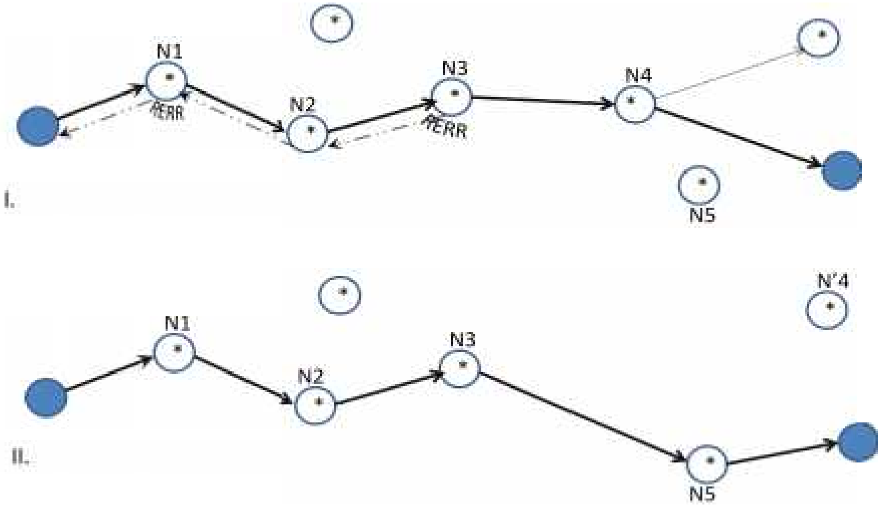

Figure 2.7: I. Route error, and II. Route maintenance

2.3.4 Management of Routing Table

A timer associated with reverse path routing entries is called the route request expiration timer. The motive of this timer is to erase, reverse route routing entries from the nodes that do not lie on the active route, and its expiration time depends on the size of the network. Moreover, in every routing table entry, the addresses of active precursors through which packets traveling to the desired destination were received are also saved. A neighbor is considered as active if it has generated or relayed a packet to the destination in question during the most recent period of the ART counter. This information is stored so that all active source nodes can be informed when a link is broken in the route to the destination. A route entry is considered to be active if it is in use by any active neighbor. The route from a source node to the destination node, followed by the packets via active route entries, is known as an active route.

It should be noted that as in DSDV [Y. Lu, *et al.*, 2003], all routes in the routing table are labeled with the sequence number of the destination, which prevents the formation of routing loops even in extreme conditions such as when nodes are extremely mobile. Each node maintains a routing table for each destination of interest. Moreover, the routing table holds various information as mentioned in section ‘2.3.1’.

If a new route is proposed to a mobile node, it compares the sequence number of the destination of the new route with the sequence number of the destination of the route it already possesses before choosing the route with the higher sequence number. If the sequence numbers are the same, the new route is only chosen when the hop count to the destination is lower.

2.3.5 Management of Local Connectivity

Nodes recognize their neighbors using two different methods. When a node receives a broadcast from a neighbor, it updates its local connectivity information to ensure that this information contains details on the neighbor in question. In some cases, where a node has not sent any packet to its active neighbors before the hello_interval has elapsed, it sends a HELLO message (a non-solicited RREP) containing its identity and sequence number [C. Perkins, 1997]. The sequence number of the node remains the same for the transmission of HELLO messages. This message cannot be diffused outside the node's immediate neighborhood as it has a TTL of one. On receiving the packet, neighbors of the source node update their local connectivity information concerning this node [C.K. Toh, 2001].

The reception of a HELLO message from a new neighbor or failure to receive consecutive HELLO messages from a node, which was previously part of the neighborhood, specifies changes in local connectivity. Inability to receive HELLO messages from inactive neighbors does not spark any reaction at the protocol level. If HELLO messages are not received from the next hop on an active path, a link rupture notification is sent to active neighbors using this next hop. This notification is carried out using RERR messages. It is considered that the maximum allowed hello loss is two.

Moreover, the management of local connectivity with hello message can also be utilized to make sure that only nodes with bidirectional connectivity are considered as a neighbor. For this purpose, each hello sent by a node lists the nodes from which it has heard. Each node verifies to make sure that it uses only routes to a neighbor that have heard the hello message of the node.

2.3.6 Advantages and Disadvantages

Advantages:

- Routes are established on demand and destination sequence numbers are used to find the latest route to the destination.
- Due to its reactive nature, it can handle highly dynamic behavior of the network.

- Lower delay for connection setup.

Disadvantage:

- Multiple RREP packets in response to a single RREQ packet can lead to heavy control overhead.
- Periodic beaconing leads to unnecessary bandwidth consumption.

2.4 Route Maintenance/Configuration Parameters

In this section, some of the default constant parameters have been talked about, which are associated with the operations of the AODV routing protocol. A particularly choice of these parameters may affect the performance of AODV protocol. This thesis work is particularly interested in the Active_Route_Timeout (ART) & Delete_Period_Constant (DPC) value. The protocol suggested that the value of ART and DPC should be a constant. In other words, their value is generalized for all kinds of applications or traffic generators. However, it has been observed in this thesis that the choice of their values according to network behavior and traffic generators may greatly increase the network performance or provide the stable routing. A particular node may wish to change the following default constant parameters as shown in the table (2.1) [C. Perkins, 1997] that is termed as the route maintenance/configuration parameters in the thesis.

Table 2.1: Various route maintenance parameters with their default value

Route maintenance parameters	**Default value**
ACTIVE_ROUTE_TIMEOUT	3 seconds
DELETE_PERIOD_CONSTANT	5
MY_ROUTE_TIMEOUT	2 * ACTIVE_ROUTE_TIMEOUT
NODE_TRAVERSAL_TIME	40 milliseconds

NET_DIAMETER	35
NET_TRAVERSAL_TIME	2*NODE_TRAVERSAL_TIME*NET_DIAMETER (Time a sender waits for a RREP)
PATH_DISCOVERY_TIME	2 * NET_TRAVERSAL_TIME
NEXT_HOP_WAIT	NODE_TRAVERSAL_TIME + 10
ALLOWED_HELLO_LOSS	2
HELLO_INTERVAL	1 second
RREQ_RETRIES	2
RREQ_RATELIMIT	10
BLACKLIST_TIMEOUT	RREQ_RETRIES *NET_TRAVERSAL_TIME
LOCAL_ADD_TTL	2
MAX_REPAIR_TTL	0.3 * NET_DIAMETER
TIMEOUT_BUFFER	2
TTL_START	1
TTL_INCREMENT	2
TTL_THRESHOLD	7

- NET_DIAMETER: It is the measurement of the network size. It can be defined as the maximum number of possible hops between any two nodes in the network. Basically, it is utilized to find out the lifetime of routes and time-out values.
- NODE_TRAVERSAL_TIME: It can be defined as the time taken by a node on average to process a packet or the average one-hop traversal time. It should contain queue, transmission, propagation, & all other delays, and its suggested default value is 40 milliseconds.

- NET_TRAVERSAL_TIME: It is the longest time that any packet may take to traverse the network (i.e. maximum waiting time that any sender can wait for RREP), and can be calculated by 2* NODE_TRAVERSAL_TIME* NET_DIAMETER.
- PATH_DISCOVERY_TIME: The maximum time that can be taken for route discovery between any two nodes in the network. It is the worst time to any network.
- NEXT_HOP_WAIT: It specifies that how long the next hop be scheduled for the processing overhead. In other words, it can be said that the time to wait before suspecting next-hop transmission of data.
- ALLOWED_HELLO_LOSS: If hello messages are in use, this parameter defines that the number of hello messages that may be lost before the route is deactivated (i.e. the maximum number of hello messages that may be missed before a neighbor is considered no longer connected).
- HELLO_INTERVAL: It defines the time interval for broadcasting hello messages.
- RREQ_RETRIES: This parameter specifies the maximum number of retransmissions of RREQs to discover a route or maximum number of RREQs sent until the destination is considered inaccessible.
- RREQ_RATELIMIT: It can be defined as the maximum number of RREQs that may be emitted per second.
- BLACKLIST_TIMEOUT: It is the maximum time duration for which node can be put into the blacklist set.
- LOCAL_ADD_TTL: It is the account for the local delays.
- MAX_REPAIR_TTL: It can be given by 0.3 * NET_DIAMETER.

- TIMEOUT_BUFFER: It incorporates a timeout for the RREP so that congestion could be counted.
- TTL_START: It specifies the TTL value whenever a RREQ message is initiated (i.e. it is the start value to expand the ring search).
- TTL_INCREMENT: It is the value by which the TTL will be incremented each time, when a request is retransmitted.
- TTL_THRESHOLD: It specifies the maximum TTL value over which NET_DIAMETER value will be used to broadcast any RREQ message.

ART, DPC and determining factors are the main concern of this thesis that has been discussed in details in further coming sections.

2.5 Active_Route_Timeout (ART)

Although reactive protocols discover routes as and when required, they still maintain route state information for a short period in its cache to avoid the overhead of route establishment. This time duration when the state information is held is called Active_Route_Timeout (ART) [C. Perkins, 1997]. In other words, ART is a period of time during which cached route is considered to be valid. When a cached route is not used for some period of time, the route state information of this cached route is removed by nodes from the routing table. The time, until the node removes the route state information about the cached route from the routing table, is called ART. After ART, cached route is considered to be expired, and its value can vary from protocol to protocol. When a route successfully establishes between two end points, it is remembered in case of its reuse in the future.

Basically, ART is a fixed parameter that tells how long route state information should be kept in the routing table after the last transmission of a packet from this route. The timer is reset back to ART whenever a route is used. The default value of this parameter is arbitrary set to 3 seconds in AODV routing, whereas it is known as Route Cache Timeout (RCT) in DSR routing and its default value is fixed at 300 seconds [C. Richard, *et al.*, 2005]. However, finding an optimal ART is not an easy task. It's a trade-off between opting a shorter ART resulting in a new route discovery process while the valid route is still there and a longer ART for sending the packets on an invalid route. The first case introduces the delay in the network that could be avoided. The consequence of the second case is the loss of one or more packets and initiation of the RERR process instead of a new route discovery process.

2.5.1 Impact of ART on Connectivity

Consider figure (2.8), where a connection between node A and node E has been established through intermediate nodes F and G after the route discovery process. At the time of route discovery process, other routes have also been searched along the destination rather than the main communication link and these are A->B->C->D->E and A->I->H->C->D->E. These other routes along the destination are known as active routes, and they are valid up to 3 seconds by default. After 3 seconds, active routes will be invalid. Suppose, during communication one intermediate node G is moved to G', then this session can be completed via other active routes without initializing the route discovery process.

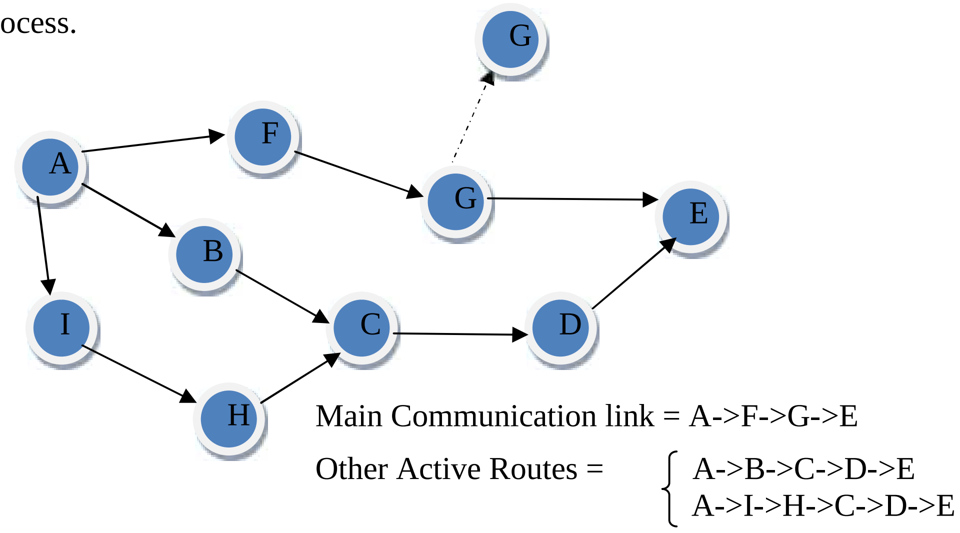

Figure 2.8: Impact of ART on connectivity

2.6 Delete_Period_Constant (DPC)

In the AODV routing protocol, the Delete_Period_Constant (DPC) denotes the multiple that determines the time after which an expired cached route is completely deleted from the routing table. An expired route is deleted after DPC multiplied by the greater of ART or hello interval.

I.e. Delete Period (DP) = DPC × Max {ART or ("Hello_Interval"=1 second)}

where suggested default value of DPC is 5.

Delete period [C. Perkins, 1997] is intended to offer an upper bound on the time for which an upstream node A can have a neighbor node B as an active next hop for the destination node D while B has invalidated the route to D as shown in figure (2.9), where C is source node and D is destination node. It is clear from figure (2.9) that if the path between node C to node D has been set up via node A and node B as intermediate node, and node B has invalidated the route to node D due to the random topology of the network then up to delete period, node B can set up the route. After the delete period, if node C still wants to communicate with node D it has to create a new setup.

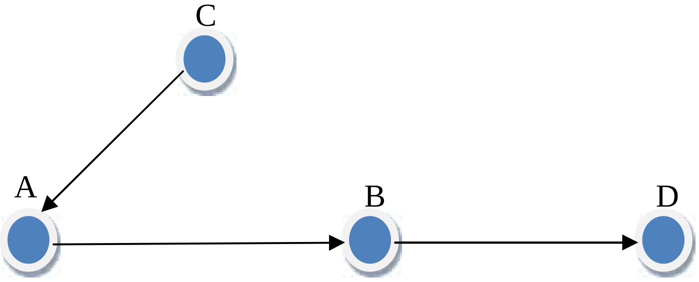

Figure 2.9: Concept of delete period

2.7 Determining Factors

Although, there are number of factors which influence the network performance in an ad-hoc network environment, this research work considers only a few factors that have a significant impact on the network performance (namely; mobility, transmission range, NLD and ANs/SD pairs). These factors are known as determining factors in this thesis work.

2.7.1 Mobility of Node

Of course, the node’s mobility has a significant effect on the performance of AODV routing in an ad-hoc network, but how it will affect the performance of routing protocol is the matter of further examination. Figure (2.10) depicts how the node’s mobility affects the connectivity of the network. In case-1, figure (2.10) demonstrates the connection between nodes X and Z through node Y. In other words, node Y plays the role of an intermediate node between these two parties where the node’s mobility is low. However, in case-2, direct connection between these two parties is possible, only if the node Z will move to Z’ position when node’s mobility is high. On the other side, a very high value of node's mobility could influence the connectivity of the node, negatively [Y. Chen, *et al.*, 2003; & K. Amjad, *et al.*, 2010]. In other words, it could be said that an immediate change in the network topology may reduce the network performance because of higher node’s mobility. In the case of lower node’s mobility, node Z has to wait for a long time in order to get the direct link, or it may choose the intermediate/relay node to get the direct link than to wait.

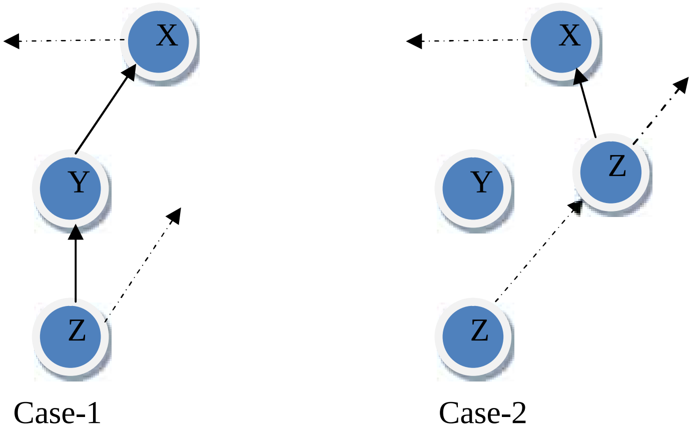

Figure 2.10: Impact of node’s mobility on connectivity

2.7.2 Transmission Range of Node

The energy conservation is a critical issue in an ad-hoc network environment because each node has limited battery power and it is not easy to replace/recharge the battery during communication [J. Tripathi, *et al.*, 2012 & 2014]. Hence, it is important to utilize the battery power efficiently to ensure longer network lifetime by utilizing the optimum power for transmission [P. Levis, *et al.*, 2009].

As, it is the well known fact that the mobile nodes of an ad-hoc network are battery operated devices. Therefore, their transmission range is limited. If the transmission power is kept high then, although all the packets have been delivered, battery power consumption will get high. If it is kept less then, although the power consumption will be low, the packets may not be able to reach the destination. In order to maximize the battery life and to get better performance, an optimum value of transmission power is to be chosen for the respective transmission ranges.

2.7.3 Network Load Density (NLD)

The Network Load Density (NLD), i.e. total number of nodes at a time in the network is also one of the paramount factors, which influence the ad-hoc network performance [Y. Chen, *et al.*, 2003]. In some cases, higher NLD within the same area gives the higher connectivity in the network. It is possible because, as NLD increases in any network, the number of relays/intermediates/interconnections between nodes towards any client would also increase. Hence, traffic congestion is decreased, and the route discovery process becomes easier. However, on the other side, within a constant area, the increase in NLD may lead heavy congestion over a network [E.M. Royer, *et al.*, 2001]. Therefore, the consequence is performance degradation in the network. So, the balance is needed between congestion and interconnection. How the NLD helps to achieve better connectivity is illustrated in figure (2.11). Consider figure (2.11), the source node P is already communicating to the destination node R via intermediate nodes Q & Z which is shown by route-2. At the same time, another source node X wants to establish a connection with node Q through route-1, where the intermediate nodes are Y and Z. In this case, there is no need to establish a connection between node Z and node Q, as it has already been established by route-2.

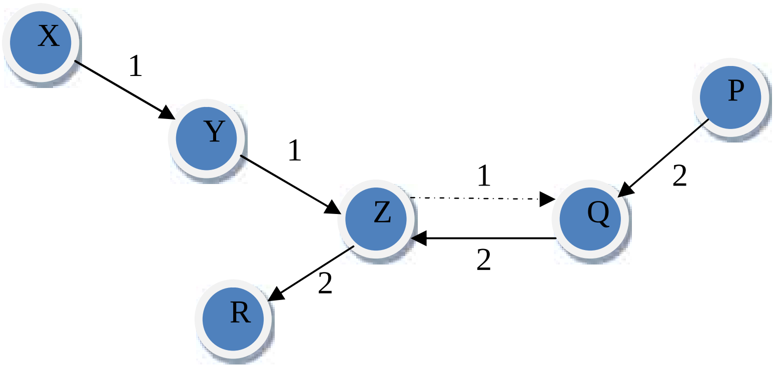

Figure 2.11: Impact of NLD on connectivity

2.7.4 ANs/SD Pair

In this thesis context, ANs stand for the number of Active Nodes/SD pairs stand for the Source-Destination pairs. Actually, this parameter defines the total number of nodes that are actively participating in communication at any instant and of course, it has a significant effect on the routing protocol performance. Therefore, research work also measures the degree at which the ANs/SD pairs will affect the QoS parameters. If ANs/SD pairs are very high within a particular area and a particular NLD then due to more number of control overhead packets, traffic congestion may also increase that may lead poor performance in the network. In other words, it can be said that the number of transmitting packets exceeds the capacity of the network that leads the network congestion. Due to this network congestion, the packets drop-rates are also increased in the network and thus, the network performance decreases.

In order to observe the best performance or to provide the stable routing in AODV network, the various route maintenance parameters are considered for the study. Hence, next coming chapter 3 clearly presents that how the AODV routing protocol's performance is influenced when various route maintenance parameters are varied from their default values. This simulation study has been conducted under the two different simulation tools.