CHAPTER 3
RESEARCH METHODOLOGY

This chapter discusses about the Dynamic Source Routing (DSR) protocol for MANET. It is divided into 3 sections. Section 3.1 explains the working of the DSR protocol. Section 3.2 presents the overall conceptual framework of the research. Section 3.3 provides the summary of the chapter.

3.1 DSR PROTOCOL FOR MULTI-HOP WIRELESS AD HOC NETWORK

DSR is a simple and efficient routing protocol designed specifically for use in multi-hop wireless ad hoc networks of mobile nodes (Johnson et al 2007). It is self-organizing and self-configuring, no pre-existing infrastructure or administration. The nodes in the network send packets to other node by allowing communication through multiple hops between nodes within the wireless transmission range of one another. Any nodes in the network may join or leave the network anytime, anywhere. The topology of the network changes continuously, so the sequence or intermediate hops needed to reach any destination may change at any time.

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 by the source node posses the absolute, ordered list of nodes through which the packet must pass through. This makes the routing to be trivially loop-free and avoiding the need for up-to-date information in the
intermediate nodes through which the packet is forwarded. The nodes which forwards this packet may easily cache its routing information for future use (Johnson et al 2001).

3.1.1 Overview of the DSR Protocol

The DSR routing protocol consist of two basic mechanisms that work together to find the route between the source and destination, and to maintain the routes in the network. They are:

- **Route Discovery**: This mechanism is used by the network when a node S wants to send a packet to a destination node D and it obtains source route to D. This mechanism is used only when S tried to send a packet to D, if the node S doesn’t know the route to D.

- **Route Maintenance**: This mechanism is used to detect the routes for S, while using a source route to D, if there is a change in the network topology, the route in the route cache, no longer uses its route to D, because the link may be broken. When the Route Maintenance indicates that a source route is broken, S can attempt to use any other routes if it knows for D else it invokes the Route Discovery process again to find a new route. This mechanism is used only when S wants to send a packet to D (Johnson et al 2001), (Johnson et al, 2007), (Samir et al 2001).

In DSR, Route Discovery and Route Maintenance each operate entirely “on demand”. In particular, unlike other protocols, DSR is a beaconless protocol, since it does not make use of hello packets to inform the neighbours about the existence of the node. Due to this on-demand behaviour and lack of periodic activity, the overhead of DSR routing protocol is reduced.
As the nodes begin to move or when the communication pattern is modified, the routing overhead of DSR automatically scales to track the routes currently in use.

DSR learn and cache multiple routes to any destination during route discovery process. So, the source node can try another cached route if the one already in use is broken. This caching of multiple routes also helps to reduce the overhead by avoiding the route discovery process each time a route in use breaks.

DSR designed to allow uni-directional links and asymmetric routes easily supported. In wireless networks, it is possible that a link may not work equally well in both the directions between the nodes, because of the propagation patterns or antenna used. DSR allows such uni-directional links to be used when necessary to improve the overall performance and network connectivity in the system.

DSR allows internetworking between heterogeneous networks, allowing a source route to be composed of hops over a combination of available networks.

3.1.2 DSR Route Discovery Process

Route Discovery aims at finding routes from a source node to the destination node in the network. When a source node wants to send a new packet addressed to a destination node, the source node places the source route in the header of the packet to be sent. The source route consists of sequence of the intermediate node that the packet has to follow on its way to destination. The source route is obtained by searching its route cache. Route cache is used to store the learned source routes. If the route is not found in its cache, it will
initiate the route discovery phase to find a new route to the destination. In this case, call the source as initiator and destination as the target of the route discovery. So, DSR routing protocol is called as Source Initiated protocol (Johnson et al 2001), (Johnson et al 2007), (Samir et al 2001).

Figure 3.1 Route Discovery example

Figure 3.1 illustrates the process of route discovery. In this example, node A wants to transmit the packet to node E. So, node A attempts to discover a route to node E. To initiate a Route Discovery, A sends the Route Request (RREQ) message to all the hops within the wireless transmission range of A including the node B. Each RREQ has the initiator, target of the route discovery and unique request identification determined by the initiator of the request. Each RREQ message encloses the record which contains list of intermediate node address through which the RREQ has been forwarded. Route record is initialized to an empty list by the initiator of the Route Discovery process. In this example node A forwards the RREQ with route record initialized with empty list.

Figure 3.2 provides the flowchart for the receipt of route request. The node which receives the RREQ, if it is the address of the target node then it returns a Route Reply (RREP) to the source along with a copy of the intermediate hops through which the RREQ is forwarded. This route is cached by the route cache of that node for further sending subsequent packet to this destination. Otherwise, if the node receiving the RREQ has already received
the RREQ with same request identification, from the same initiator to the
destination then the packet is already received by the node, or if this node’s
address is already available in the route record then it discards the request.
Otherwise, this node appends its own address to the route record in the RREQ
and propagates it to the next hop by means of using broadcasting. In the given
example, node B broadcast the RREQ message which is received by C; in turn
node C and D broadcast the RREQ, resulting in receipt of a copy of the
request by the node E.

Figure 3.2 Receipt of Route Request
In returning Route Reply (RREP) to the initiator of the route discovery process, i.e., from the given example A is the initiator and E is the target, after receiving the RREQ packet, node E wants to send the RREP packet back to A.

Figure 3.3 provides the flowchart for the Route Reply mechanism. From the figure 3.1 node E initially examines its own route cache for a route back to A. If the route is found, then it will use it as the source route to deliver the packet containing the Route Reply. Otherwise, node E may perform its own Route Discovery process for the node A, but to avoid the recursive calling of Route Discovery, it piggybacks the route reply on its own route request message for A.

Figure 3.3 Route Reply mechanism
Node E simply reverses the series of hops in the route record that it tries to send in the route reply packet. Node E uses this address as the source address on the packet which carries the route reply itself. For MAC protocol such as IEEE 802.11 require a bi-directional frame exchange as a part of the MAC protocol, this route reversal is used to avoid the overhead caused by the route discovery and also checks the discovered route is bi-directional or not before the initiator of the route discovery process uses this route. So, it prevents the discovery of routes using the uni-directional links.

3.1.3 DSR Route Maintenance Process

Many wireless networks make use of hop-by-hop acknowledgement at the data link layer in order to predict the detection and retransmission of vanished or polluted packets. In this route maintenance phase, it determines whether the nodes in the source route are still working or not. If the data link layer reports a transmission problem for which it cannot recover, the node send a route error packet to the initiator to encounter the error. The route error packet contains the address of the nodes at both ends of the node ie., the node that detect the error and the node which tries to transmit the packet. When the route error packet is received by the initiator, then it removes the route from the route cache (Johnson et al 2001) (Johnson et al 2007) (Samir et al 2001).

![Figure 3.4 Route Maintenance example](image)
From the figure 3.4 node A initiates a packet for E using the source route in the route cache through the intermediate nodes B, C, D. In this case the node A is responsible for receiving the packet at B, B and C is responsible for receiving the packet at C and D, D is responsible for receiving the packet finally at the destination node E. This confirmation may be offered at no cost to DSR, either as existing standard part of the MAC protocol in use (since the link-level acknowledgement frame is defined by IEEE 802.11) or by a passive acknowledgement. If neither of these confirmations is received, the node transmitting the packet may set a bit in the packet’s header to request a DSR-specific software acknowledgement be returned by the next hop. This software directly sends the acknowledgement to the initiator. If the link between these nodes is uni-directional, this software acknowledgement can travel over a different and multi-hop path.

If this confirmation is not received after some maximum number of retransmission of packets, this node returns Route Error message to the initiator of the packet, spotting the link over which the packet could not be successfully transmitted. In the figure 3.4, node C is not able to forward the packet to node D, after trying for several times, C returns a Route Error message to A, informing that the link between C and D is currently broken.

3.1.4 Additional Route Discovery Features

- Caching Overhead Routing Information: Node which is forwarding or overhearing any packet can add the routing information from that packet to the route cache of the node. The node may cache the source route using either the accumulated route record from a Route Request or the route being sent in a Route Reply. Routing information from any of these packets received can be cached, whether or not the packet was addressed to
the present node, sent to a broadcast MAC address, or received whereas the node’s network interface is in promiscuous mode.

The limitation of caching of overhead routing information is the possible presence of the uni-directional links in the ad hoc network (Johnson et al 2001), (Johnson et al 2007), (Samir et al 2001).

- **Replying to Route Request using Cached Routes:** Whenever a node receiving the Route Request for which it is not the target, it searches its Route Cache for a route to the target of the request. If the route is found, the node returns the Route Reply to the source itself rather than forwarding the Route Request packet. In the Route Reply, it adds the sequence of nodes through which the Route Request was forwarded to it, and also appends the route from itself to the target from its Route Cache.

However, before transmitting a Route Reply which was created using the information from its Route Cache and after concatenation it must verify the resulting route list to see if it contains any duplicate nodes in the route record.

- **Preventing Route Reply Storm:** If a node broadcasts a Route Request for a destination node, if the node’s neighbour’s has the route for the destination in their route cache, each neighbour attempt to send a Route Reply at the same time, since they all receive the Route Request packet at the same time. Such concurrent replies from different nodes all receiving the Route Request may create collisions among some or all of the replies which lead to the network collision in the area. This wastes the bandwidth and increases the load in the network.
If a node puts its network interface into the promiscuous receive mode, it should delay sending its own Route Reply for a short route first. During the delay, the hop receives and can examine all data packets from the source and can infer whether the initiator of the Route Discovery has already received a Route Reply showing a better route.

- **Route Request Hop Limits**: This mechanism is used to determine whether the target is neighbour of a source node or if the neighbours contain the route for the target cached. In this mechanism, Route Request packet contains the “hop limit” which is used to restrict the number of intermediate nodes allowed to transmit the copy of the Route Request. Whenever the Route Request packet is forwarded by each node, then the hop limit is reduced by one, and the Route Request packet is discarded from the network when the value of hop limit reaches zero before finding the destination.

3.1.5 Additional Route Maintenance Features

- **Packet Salvaging**: Node after sending a Route Error message as a part of Route Maintenance, it tries to salvage the data packet that causes the Route Error rather than removing it from the network. In order to salvage a packet the node sending a Route Error searches its route cache for an alternate path from itself to the destination. If the route is found, the node may salvage the packet after returning the Route Error by replacing the original source route from the packet by the route from its route cache. The node then transmits the packet to the next node along with this source route.

When salvaging a packet by this method, the packet is marked as salvaged in order to save the packet from salvaging several times. Otherwise,
it is possible for the packet to enter a routing loop when different nodes repeatedly salvage the packet and changes the source route on the packet with routes to each other. Alternative mechanism of salvaging is to change only the unused suffix of the original route with the route from the corresponding node’s Route Cache, forms a new route whose suffix is from the route cache of the node and prefix is from the initiator i.e., original route. The concept of salvaging prevents the new route from backtracking from this node to a previous nodes already travelled by this packet, to then be forwarded along a different remaining sequence of hops to the destination. This salvaging allows backtracking but won’t allow a packet from being salvage more than once (Johnson et al 2001), (Johnson et al 2007), (Samir et al 2001).

- **Automatic Route Shortening:** In this automatic route shortening mechanism, Source routes in the cache may be automatically shortened if one or more intermediate nodes in the route no longer needed for transmitting the packets. This approach of automatic shortening routes in use is somewhat similar to the use of passive acknowledgements. If a node by operating its network interface in promiscuous receive mode, is able to overhear a packet carrying a source route. Then this node examines the unused portion of the source route. The intermediate nodes in the source route are not needed if the node is not the intended next node for the packet but is named in the later unused portion of the source route of the packet (Johnson et al 2001), (Johnson et al 2007), (Samir et al 2001).

- **Increased Spreading of Route Error Messages:** When a initiator of the packet receives the route error for a data packet that it originated, this source hop propagates this route error to its neighbour by piggybacking it on its next Route Request. In this way, stale information in the route cache of nodes around this initiator of the packet will not produce
Route Replies that has the same invalid route for which the source node received the Route Error (Johnson et al 2001), (Johnson et al 2007), (Samir et al 2001).

- **Caching Negative Information**: DSR routing protocol can potentially advantage from nodes caching “negative” information in their route caches (Johnson et al 2001), (Johnson et al 2007).

### 3.1.6 Route Cache in DSR Routing Protocol

Route cache strategy in DSR routing protocol was proposed by Johnson et al in 1998. Route caching is the most important approach to reduce the flooding of the network by neglecting the route discovery process as much as possible (Husieen et al 2011), (Marina & Das 2001), (Lou & Fang 2002). Route cache is an essential component in DSR protocol. Route cache is used to store the routes that have been learnt from the source route in order to avoid unnecessary route discovery operation when it wants to transmit a data packet. The reinitiating of a route discovery in the reactive routing protocol is costlier in terms of energy, delay and bandwidth. So, it successfully reduces the overhead and the delay time. Due to high mobility and high traffic load in the network leads to the breakage of link. While on the other side, the route reply storm and the stale links produced by route cache strategy causes the long delay, maximize the packet loss, overhead which degrade the performance of the DSR routing protocol (Chen et al 2010).

The effect of route cache in the performance of DSR was studied by Maltz et al. DSR routing protocol does not have an efficient mechanism to find the fresh route from the stale routes. Route cache is of two types. They
are path cache and link cache (Sharma 2008), (Shoba & Rajanikanth 2009). Figure 3.5 & Figure 3.6 gives the organization of path and link cache.

**Path Cache:**

When every source node receives Route Reply, it consists of complete source route in the sequence of link between them. It is very simple to implement and it helps to identify that the path is loop free. Searching for any destination is also simple.

![Figure 3.5 Path Cache Organization](image)

**Link Cache**

In the case of link cache, source node adds every link returned by Route Reply to a topology graph. In this cache, searching is very difficult compared to path cache.

![Figure 3.6 Link Cache Organization](image)
Use of Route caching

Route Cache in the mobile node is used to speed up of route discovery process and in the reduction of propagation of route requests.

Beware of Route Caching

- Stale caches can adversely affect the performance of DSR
- With passage of time and host mobility, cached routes may become invalid
- All cached routes containing a failed link are not erased by route error
- A sender node may try several stale routes (obtained from local cache or replies from cache of other intermediate nodes) before finding a valid route

3.1.7 Issues in DSR Route Cache

DSR does not have any effective mechanism to reduce the stale cache problem. Route cache in each node stores multiple routes that have been found via., flooding or through route replies from the intermediate node which may possess stale routes. Data transmission through stale route leads to the problem of increase in overhead, and polluting other caches. Some of the factors that create the stale route and the outcome of it are provided below:-

- Node mobility: In case of high node mobility, entries in route caches hastily become invalid or ineffective. Data packets suffer needless delays, when an ineffective route is used for transmission. Route failures will generate flooding, creates supplementary latency for data packets, when an invalid route is followed (Khetrapal 2006).
- **Promiscuous overhearing:** Stale routes will be rapidly propagated to caches of other nodes because of the use of responding to the route request RREQ with cached routes. Hence, pre-active and post-active routes are the key sources of cache staleness (Yu 2006).

- **Incomplete error notification:** In case of link breakages, the routes having an access with the busted link are not propagated to all caches. As an alternative, the route error is unicasting merely to the source, whose data packet is responsible for identifying the link breakage by means of a link layer feedback. Therefore only a limited number of caches are cleaned (Marina & Das 2001).

- **Expiry:** Currently, there is no effective mechanism to expire stale routes. If they are not cleaned explicitly by the error mechanism, the stale cache entries will reside permanently in the cache (Marina & Das 2001).

- **Quick pollution:** There is no technique to resolve the freshness of any route information. For instance, even if a route error removes the stale cache entry, the succeeding routed data packet carrying the same stale route can set that entry right back in. Therefore, cache pollution can transmit quite rapidly (Marina & Das 2001).

### 3.1.8 Advantages and Disadvantages of DSR

This section provides the number of advantages and disadvantages of the DSR routing protocol (Samir et al 2001), (Lin 2004), (Marina & Das 2001).

- **Advantages of DSR**

  1. **On demand routing:** Routes maintained only between nodes that
need to communicate.

2. Route Caching: This reduces the overhead of route maintenance.
3. Route caching can further reduce route discovery overhead.
4. A single route discovery may yield many routes to the destination due to intermediate node replying from local caches useful when route breaks.
5. Intermediate nodes make use of the route cache information efficiently to reduce the control overhead
6. No periodic routing of packets are required which reduces the utilization of resource.

Disadvantages of DSR

1. Not Scalable: Packet header size grows with route length due to source routing and DSR relies on blind broadcast to discover routes.
2. Network-wide flood: Flood of route request may reach all nodes in the network.
3. Route maintenance mechanism does not locally repair a broken link.
4. Collision: Care must be taken to avoid collisions between route requests propagated by neighbouring nodes.
5. Route reply storm problem- increased contention if too many route replies come back due to nodes replying using their local cache.
6. Stale cache problem: An intermediate node may send route reply using a stale cached route thus polluting other caches.
7. Connection setup delay is higher than that of table-driven protocols. Causes long delay before the first data packet sent.
8. Performance of the routing protocol degrades rapidly with increased mobility.

3.2 CONCEPTUAL FRAMEWORK OF THE RESEARCH

This section illustrates the overall framework of the research. Figure 3.7 gives the overall frame work of the research. This research is concerned with Mobile Ad Hoc Network. Due to the dynamic topology characteristic of MANET, routing is a challenging task. Number of routing protocols was designed based on the various characteristics. DSR is a prominent, and reactive on-demand routing protocol. DSR aggressively makes use of route cache, each node in the MANET updates its route cache based on the requirement of the route i.e. on-demand basis.

Due to its dynamic topology, the performance of the DSR is affected by the various factors such as stale route problem, traffic load and scalability. In this research, three techniques are proposed to obtain reliability and optimize the DSR routing protocol.

Reliable and Effective Cache Management technique is proposed to attain the reliability of routing by considering the link quality. This reduces the stale cache problem occurred in the route cache which degrade the performance of the DSR routing protocol. In this technique, the quality of the link is measured using the Combine Weight Function (CWF). The CWF is calculated using the signal strength of the link, traffic load across the route, energy consumption of the route and length of the route. Based on the CWF the routes are arranged with maximum energy and signal strength with minimum load and length. The routes in the route cache with minimum combined weight function are removed from the list, when the link is likely to be broken.
Figure 3.7 Overall Framework
Reliable and Effective Load Balanced technique is designed to achieve the reliability by distributing the load over the multiple paths. The multiple paths are selected by means of using the combined weight function. The path with minimum combined weight function implies that no information reaches the destination. The data sent through the path with greater than minimum combined weight function reaches the destination. Then the traffic is distributed over these paths using the network diversity coding. The original message is reconstructed on the destination.

Optimized Reliable and Load Balanced Routing Protocol is designed to optimize the reliable and effective load balanced routing protocol by reducing the overhead, which increases the scalability of the routing protocol. The high transmission overhead is involved due to the header size of the packet which comprises the IP addresses of all the nodes involved in the transmission. In this technique, the source route in the packet is compressed using the bloom filter function. It deletes the IP addresses of the node already visited before forwarding to the next node. This prevents the packet size of the data to the original size of the data without overloading it. This reduces the overhead and hence the scalability is increased.

3.3 CHAPTER SUMMARY

The chapter discusses about the DSR on-demand routing protocol. Dynamic Source Routing protocol (DSR) provides outstanding performance for routing in multi-hop wireless routing protocol. From the above study, it shows that DSR has very low routing overhead and also it is able to provide almost all the originated data packets during the mobility of the network.
The key reason for providing good performance is that it operates entirely on an on-demand basis and does not need periodic activity within the network. So, DSR is called beaconless routing protocol. In this chapter the key mechanism of Route Discovery and Route Maintenance are studied. It shows that how it enables wireless mobile nodes to automatically form a completely infrastructure less network. Since, it is beaconless protocol, the routes in the route cache becomes stale due to changes in the topology. Whenever the nodes in the network increase, due to source routing, the transport overhead also increases, which affects the DSR routing protocol.

It also provides the overall framework for the proposed work. In the forthcoming chapters it explains the techniques given in this chapter.