**3**

**DYNAMIC SOURCE ROUTING PROTOCOL(DSR)**



**Soha Muskaan Sayyad -AP20110010602**

**Gowthami Gurram-AP20110010571**

**Mounica Noti-AP20110010575**

**Purnima Singh-AP20110010590**

**Meghana Amara-AP20110010577**

**Jaya bhavani-AP20110010567**

**Pallavi Annapareddy AP20110010564**

**Gayathri Kondaveeti-AP20110010568**

**INTRODUCTION:**

Dynamic Source Routing Protocol comes under the Reactive Routing protocol.It finds the path only when required or basically on demand it will find the path to send the packet from source to destination. It uses the “Route Discovery Mechanism” to find the path from source node to destination node using the intermediate nodes.It uses “Re-Request” packet to inform the source that the packet have reached the destination through the shortest path.This algorithm also does the “Route Maintenance” generally when a path or link is broken.

**OBJECTIVES:**

* To show how the DSR algorithm works.
* To show the result using an example graph.
* To show the result of the analysis

**DSR ALGORITHM:**

**Route Discovery:**

The source sends a broadcast packet which contains source address, destination,request id and path. If a host saw the packet before,discards it. Otherwise, the route looks up its route caches to look for a route to destination.If not found, appends its address into the packet, rebroadcast. If it finds a route in its route cache, it sends a route reply packet, which is sent to the source by router cache or the route discovery.

Generally this algorithm has few steps. It starts finding the path from the source destination only when the request for sending packet is arised.

**Example 1: Graph that is used for analysis**

Enter number of nodes in the graph:5

Enter the adjacency matrix:

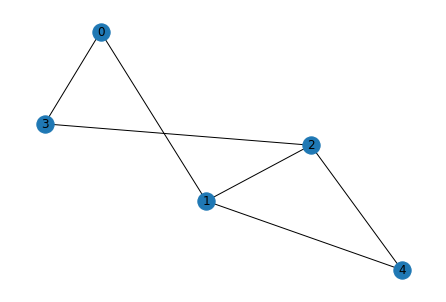
Enter the row : 0 1 0 1 0

Enter the row : 1 0 1 0 1

Enter the row : 0 1 0 1 1

Enter the row : 1 0 1 0 0

Enter the row : 0 1 1 0 0



**Step 1:** In this let we select the source and the destination node. The unique id of every node is already given.

**Step 2:** Source node will flood the request to the destination node via all the nodes.

**Source**

| 0 |
| --- |

**Destination**

| 4 |
| --- |

Let us consider the unique id as 4 for source node 0

Now we need to find all possible paths from source to destination.

Initially let's start from 0 which is the source node.

**Step 3:** First we have two ways from 0 those are

1. 0—>1
2. 0—>3

Let's explore 1

**Step4:** From 0→1 we have two ways those are

1. 0→1→4
2. 0→1→2

Let’s explore 1

**Step 5:** From 0→1→4 we have reached the destination path which is 4

**Path 1**

| 0→1→4 |
| --- |

**Step 6:** Now let’s explore 2 nd path from 0→1 i.e is 0→1→2.

From 0→1→2 we have one path i.e is

0→1→2→4 Here we have reached the final destination node which is node 4 hence this path is completed.

**Path 2**

| 0→1→2→4 |
| --- |

**Step 7:** Now let’s explore the second path from source which is 0→3 . From this we have one path.

1. 0→3→2.

Now let’s explore 1.

**Step 8:** From 0→3→2 we can reach the goal node in two ways either directly or via node 1.

1. 0→3→2→4
2. 0→3→2→1

Now let’s explore 1

**Step 9:** The path 0→3→2→4 is the path that is reaching to goal node 4 .

**Path 3**

| 0→3→2→4 |
| --- |

Now let’s explore 2nd path from 0→3→2

Step 10: From path 0→3→2→1 we can reach directly to goal node

**Path 4**

| 0→3→2→1→4 |
| --- |

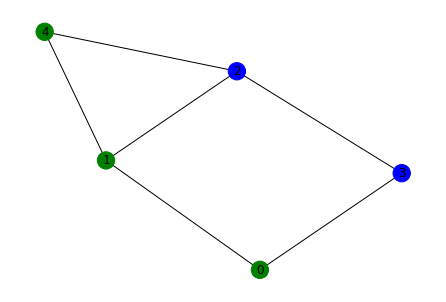
**Step 10**: we have got all possible paths from source to destination. Now the destination sends the ‘Reply Packet’ to the source node.

**Reply Packet:**

If the destination has a route to the source in its route cache, use it.Else if symmetric links are supported, use the reverse of the router record. Else if symmetric links are not supported, the destination initiates route discovery to source.

Generally the reply packet is sent to the source from the shortest path from destination to source or vice versa. This path is taken from the list of the paths that was found above during flooding the request.

**Path for Reply packet sending:**



**Step 1**: Find the shortest path from all the paths.

path from src 0 to dst 4 are

0 1 4

0 1 2 4

0 3 2 4

0 3 2 1 4

From above the path 0→1→4 is the shortest path hence from this path the reply packet is sent.



**Example 2:**

**Step 1:** It finds which is the source node and which is destination node and the unique id . In the given graph A is the source from where we started and E is where we want to go (destination).Here 4 is unique id

**Step 2:** This is the initial packet which we are transmitting the path we will send this packet to the all nodes connected so here the connected nodes with A are B and C 4AE A→B is a path from A to B like similarly 4AE A→C so we will be checking with nodes B and C to D we will get path 4AE A→B→D from C 4AE A→C→D in this way path will be updated in every step.

**Step 3:** Both the packets are not accepted at node D and one packet will be sent back. Let us assume the packet is sent from D to C and from A to C to E the path is 4AE A→C→E.

**Route Maintenance:**

This is also taken care of by the algorithm whenever a node is added to the network or when an existing node is removed from the network. There may be some times a path or link may be broken between the nodes at which time this algorithm finds a new path. This is basically called Route Maintenance.

**Algorithm Implementat**

This algorithm is basically constructed by using the help of the Breadth First Search algorithm. BFS generally uses level wise traversal. By using this algorithm we can find all possible paths from the source to destination.Hence to find all possible paths we have used BFS. Generally to find the shortest path, we can use the Dijkstra algorithm but in our code we have just taken the shortest path from all the paths found using BFS.

#### **Advantages :** Dynamic Source Routing Protocol

* A perfect route is always discovered.
* Highly efficient.
* Low bandwidth Consumption.

#### **Disadvantages :** Dynamic Source Routing Protocol

* If the route gets broken, data transmission cannot happen.
* Time taking algorithm-Slow.
* If the network is large , then it is impossible for the data packet header to hold the whole information of the routes.

**Conclusion:**

Excellent performance for routing in multi-hop ad hoc. Very low routing overhead. Able to deliver almost all originated data packets, even with rapid motion of all nodes.