**DSR Test Cases**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Description** | **Expected Results** | **Actual Results** | **Comments** |
| **Route Discovery** | Automatic route discovery | DSR successfully performs automatic route discovery. | PASS |  |
| On-Demand route discovery | DSR successfully performs route discovery only on demand. No periodic broadcasting involved. | PASS |  |
| **Route Request** | Sending route request | DSR successfully allows nodes firstly search their own route cache to see whether they have stored route to destination node  If nodes have the route, then nodes successfully sends that route to source node, else, nodes broadcast the route request message to their neighbours and attach their own ID to the route request message. | PASS |  |
| Route request frequency | DSR successfully allows source node to wait for a fixed amount of time before initiating another route request to avoid flooding the network. | PASS |  |
| Hop limit | DSR successfully controls the route request to neighbourhood area to avoid unnecessary route reply message. | PASS | The simulator defined a default communication matrix to determine which node can talk to each other at each time instance. This avoids unnecessary route reply message. |
| Restricted Propagation of Route Request | DSR successfully increases hop count proportionately when receiving no route reply messages to avoid network congestion an unnecessary route reply messages (i.e. start with distance 2, then when no reply message, increase distance to 3 and so on). | FAIL |  |
| **Route Cache** | Maintaining route cache information | Each node successfully maintains a route cache and remember the routes that it has learnt about | N/A | Has not being implemented |
| Route cache updates | DSR successfully maintains an up-to-date information in the route cache for each node | N/A | Has not being implemented |
| Route cache has no available information | DSR successfully initiates a new route discovery when no route cache information available. | N/A | Has not been implemented. |
| Caching overhead routing information | DSR successfully allows node P to store overhear routing information in its route cache from node B to node C. DSR is also successfully use this information when node P receive route request from node B to node C. | N/A | Has not been implemented. |
| **Route Reply** | Complete route discovery | DSR successfully sends route reply message from destination node that has route to destination in its route cache. | FAIL | Route cache has not been implemented yet and the nodes are generated randomly. |
| Partial route discovery | DSR successfully sends route reply message from intermediate node that has route to destination in its route cache | FAIL | Route cache has not been implemented yet. |
| Waiting before reply | All nodes successfully wait for a random amount of time and listen to the traffic before sending route reply message to avoid network congestion and packet collisions. | FAIL |  |
| Accumulated route reply | DSR successfully sends route reply message back to the source node using accumulated route (the nodes through which it has passed). | N/A |  |
| Route reply storm | DSR successfully avoids route reply storm (many nodes try to send route reply for the same destination which may flood the network). | FAIL |  |
| **Route Maintenance** | Alternative route for broken route | DSR successfully uses alternative route stored in route cache when priority route is broken. | N/A | Has not being implemented |
| New route discovery for broken route | DSR successfully discovers new route when priority route is broken and route cache has no alternative route stored. | N/A | Has not being implemented |
| Active acknowledgment | DSR successfully retransmits packet for a fixed number of times if no acknowledgement received. | PASS |  |
| Passive acknowledgement | DSR successfully sends acknowledgement to node A when node A overhear the forwarding of the packet to node B and knows that node B successfully received the packet. | PASS |  |
| Spreading route Error Message | DSR successfully sends error message if the nodes do not receive any acknowledgement after retransmit message for a fixed number of times. |  |  |
| On-Demand route maintenance | DSR successfully operates route maintenance only on demand. No periodic broadcasting involved. |  |  |
| Changing in communication pattern | Number of overhead packets increases and DSR is successfully performs new route discovery and new route discovery packets are the overhead packets. |  |  |
| Packet salvaging | DSR successfully indicates to the other nodes that the packet sent has been salvaged when the node receives route error message and re-send the packet that cause route error. |  |  |
| Automatic route shortening | DSR successfully performs automatic route shortening.  Destination node informs source node that it can ignore several intermediate nodes |  |  |
| **Others** | Packet Size | DSR successfully sends smaller size of overhead packets (DSR is designed to send smaller packet). | PASS |  |
| Drop packets when nodes are static | DSR successfully drops the number of overhead packets to zero when the nodes are static and all routes have been discovered. |  |  |
| Energy-efficient | DSR successfully shows more energy-efficient and does not congest the network with too many control messages. |  |  |
| Unique ID | DSR successfully assigns unique ID for each node. | PASS |  |
| Promiscuous mode of operation | DSR successfully allows each node to overhear or not to overhear other nodes' transmission. |  |  |