CHAPTER 2

MOBILE AD HOC ROUTING USING
INFORMATION SYSTEMS

2.1 DYNAMIC SOURCE ROUTING

This chapter introduces the use of information systems of Rough Set Theory in routing in mobile ad hoc networks (Rajam et al. 2006). The different routing algorithms proposed in this thesis have modified the Dynamic Source Routing (DSR) protocol to incorporate the notion of information systems. The working of the DSR protocol is explained in detail in this section (Johnson and Maltz 1996).

2.1.1 Overview

When a mobile node known as the source node wants to send a data packet to another node, it constructs a source route in the data packet’s header. The source route has the address of each mobile node through which the packet has to be forwarded to reach the destination node. The source node now transmits the packet to the next node according to the source route. Each mobile node then forwards the packet to the next node, known as the next hop, according to the source route in the packet, till the packet reaches the destination.

Each mobile node has a route cache in which it caches source routes
that it has learnt. When a source node sends a packet to a destination node, it first checks in its route cache for a source route to the destination. If there is a source route, this route is used as the source route in the data packet’s header. If there is no route to the destination in the route cache, then the node discovers a route using the route discovery procedure.

While a node is using any source route, it monitors the correct operation of the route. That is, if any of the mobile nodes in the source route has moved away from the transmission range of its next hop mobile node, then the route is no more correct. This monitoring of the correct operation of the route is called route maintenance.

The basic operations of route discovery, route maintenance and the optimisations that have been done to the DSR protocol are explained in the following subsections.

2.1.2 Route discovery

The route discovery procedure is used to find the route from any node to any other node that is a hop or multiple hops away in the ad hoc network. A mobile node initiating a route discovery broadcasts a route request (RREQ) packet that is received by all the nodes within its transmission range. The RREQ packet has the initiator’s address, the address of the node for which a route is to be found, that is, target’s address and a route record. In the route record is accumulated a record of the sequence of hops that the RREQ takes as it is propagated through the network during the route discovery. The RREQ packet also contains a unique request id, set by the initiator of the RREQ. Each mobile node maintains a list of the (initiator address, request id) pairs that it has received
on any route request.

When a node receives a RREQ packet, it processes the packet according to the following steps:

1. If the pair (initiator address, request id) for this RREQ packet is present in the list maintained in the node, then this packet is discarded.

2. Otherwise, if this node’s address is already present in the route record of the RREQ packet, then it is discarded.

3. Otherwise, if the address of this node matches the target’s address, the route in the route record has the route from the initiator to this node. A copy of this route is sent to the initiator of the RREQ in a route reply (RREP) packet.

4. Otherwise, the address of this node is appended to the route record and the RREQ packet is re-broadcasted.

So, finally when the RREQ reaches the destination, a RREP is sent back to the source node.

2.1.3 Route maintenance

When a mobile node, while transmitting a data packet to the next hop finds that the transmission is not successful, it sends a route error (RERR) packet to the source of the data packet. The RERR packet has the addresses of
both ends of the hop in error: the node that detected the error and the node to which it was attempting the transmit the packet.

When the RERR packet is received by a node, the hop in error is removed from the route cache of the node and all routes which contain this hop are truncated at this point.

2.1.4 Optimisations

A node can add entries to its route cache whenever it learns a new route. A node can learn a new route from the source route of a data packet or from the route record of a route request packet or from the route in a route reply packet that it forwards.

A node can avoid propagating a route request if it has a route to the target node. If a node that receives a RREQ packet finds that there is a route to the target node in its route cache, it appends the cached route to the route record in the RREQ, puts this route in a RREP packet and sends it back to the source node. The node does not re-broadcast the RREQ packet. But this concatenation of the route from the route cache to the route record in the RREQ, may sometimes result in a loop in the new route formed in the route record. In this case, this node discards the request if the route in its reply would contain a loop.

To avoid propagation of many redundant copies of the route requests, the initiator first sends the route request with a hop limit of one. If no route reply is received from this route request after a small timeout period, a new route request with the hop limit set to a predefined maximum value is sent.
2.2 INFORMATION SYSTEMS AND DECISION SYSTEMS

2.2.1 Information system

In Rough Set Theory, a data set is represented as a table, where each row represents an *element* (or an event or an object or an example or an entity). Each column represents an *attribute* that can be measured for an element. This table is called an *information system*. The set of all elements is known as the *universe*.

Consider a universe $U$ of elements. Formally, an information system $I$ is a quadruple $I = (U, A, V, \rho)$, where

- $A$ is a non-empty, finite set of attributes;
- $V = \bigcup_{a \in A} V_a$ is the set of attribute values of all attributes,
  where $V_a$ is the set of possible values of attribute $a$;
- $\rho : U \times A \to V$ is an information function,
  such that for every element $x \in U$,
  $\rho(x, a) \in V_a$ is the value of attribute $a$ for element $x$.

The information system can also be viewed as an *information table*, where each element $x \in U$ corresponds to a row, and each attribute $a \in A$ corresponds to a column.

$I = (U, A \cup \{d\}, V, \rho)$, is known as a *decision system*, when an attribute $d$ is specified as the *decision attribute*. A decision system is used for predicting the value of the decision attribute. $A$ is then known as the set of *condition attributes*.

For example, if the information system describes a hospital, the
elements may be patients; the condition attributes may be symptoms and tests; and decision attribute may be diseases.

2.2.2 Indiscernibility

In an information system, elements (objects) that exhibit the same information are indiscernible and are called elementary sets. Subsets of the set of all elements (objects) with the same value of decision are called concepts. The concept \( X \subseteq U \) is the set of elements of \( U \) that have a particular value (say, \( t \)) of the decision attribute \( d \). That is, \( X = \{ x \in U \mid \rho(x,d) = t \} \).

A binary relation \( R \subseteq X \times X \) which is reflexive, symmetric and transitive is called an equivalence relation. The equivalence class of an element \( x \in X \) consists of all objects \( y \in X \) such that \( xRy \). An equivalence relation \( R \), called indiscernibility relation, is defined on the universe \( U \) as

\[
R = \{ (x,y) \in U \times U \mid \forall a \in A, \rho(x,a) = \rho(y,a) \} \quad (2.1)
\]

In the information system \( I \), the elementary set containing the element \( x \in U \), with respect to the indiscernibility relation \( R \), is

\[
[x]_R = \{ y \in U \mid yRx \} \quad (2.2)
\]

2.2.3 Set approximation

For each concept \( X \), the greatest definable set contained in \( X \) is called the lower approximation of \( X \) and the least definable set containing \( X \) is called
the upper approximation of \( X \). The set containing the elements from the upper approximation of \( X \) that are not members of the lower approximation of \( X \), is called the boundary region. A set is said to be rough if the boundary region is non-empty. A set is said to be crisp if the boundary region is empty.

The lower approximation of the concept \( X \subseteq U \), with respect to \( U \) and equivalence relation \( R \) on \( U \), is the union of the elementary sets of \( U \) with respect to \( R \) that are contained in \( X \), and is denoted as \( \overline{RX} = \{ x \mid [x]_R \subseteq X \} \).

The upper approximation of \( X \) is the union of the elementary sets of \( U \) with respect to \( R \) that have a non-zero intersection with \( X \), and is denoted as \( \overline{RX} = \{ x \mid [x]_R \cap X \neq \emptyset \} \).

The lower approximation of \( X \) is also known as the Positive region of \( X \). The set \( BN_R(X) = \overline{RX} - RX \) is called the Boundary region of \( X \). The set \( U - \overline{RX} \) is called the Negative region of \( X \).

### 2.3 INFORMATION SYSTEM AND DECISION SYSTEM IN A MOBILE NODE

#### 2.3.1 Information system

Let \( M \) be a set of mobile nodes. A route is a path through mobile nodes in \( M \) and is denoted as a sequence of mobile nodes \( m_1m_2\ldots m_k \), \( m_i \in M, i = 1, \ldots, k \). Each mobile node \( m \in M \) maintains a route cache that stores all the routes that \( m \) knows. Any route in the route cache is a path starting from that mobile node \( m \), and so \( m_1 \), the first node in the route, is \( m \) itself. Any route in the route cache is a simple path, where no node repeats, that is, \( m_i \neq m_j \) for
$m_i, m_j$ in the path, $i \neq j$. So, $k \leq |M|$.

Each mobile node $m \in M$ has an information table $I_m = (U_m, A_m, V_m, \rho_m)$ associated with it. Each row ($x \in U_m$) in the information table $I_m$ corresponds to a route in the route cache maintained by that mobile node $m$.

Let the set of condition attributes $A_m$, correspond to the set of all mobile nodes. That is, each condition attribute $a \in A_m$ corresponds to a mobile node $m' \in M$. So, $A_m$ is the same for any $m \in M$ and is denoted as $A$. That is, $A = M$. Hence, the number of condition attributes, is equal to the total number of mobile nodes. Each condition attribute is a boolean attribute, with the value of the attribute $V_a = \{0, 1\}$, and is set to 1 or 0 depending on whether that mobile node ($m \in M$) corresponding to $a \in A$ exists in the route corresponding to that element. So, since $V_a = \{0, 1\}$ for every $a \in A$, $V_m = \{0, 1\}$ and is denoted as $V$ since it is the same in each mobile node $m$.

Consider an element $x \in U_m$ corresponding to a route $m_1m_2 \ldots m_k$, $m_i \in M, i = 1, \ldots, k$. $\rho_m(x, m_i) = 1$, for each $m_i$ in the route, and is 0 for all other $m \in A$. That is, when the row is added to the information table, the values of the condition attributes corresponding to the nodes $m_1, m_2, \ldots, m_k$ are set as 1.

When a route $m m_1m_2 \ldots m_k$, $m_i \in M, i = 1, \ldots, k$ is to be added to the route cache, a row corresponding to this route is always added to the information table. However, with respect to the actual route cache itself, the route is added only if it is not identical to any route already existing in the route cache or a subpath of any other route already in the route cache.

Consider a mobile node $m$ that learns the routes $m m_1m_2m_4m_5$, 


Table 2.1: Route cache

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$mm_1m_2m_4m_5$</td>
<td></td>
</tr>
<tr>
<td>$mm_1m_2m_5$</td>
<td></td>
</tr>
<tr>
<td>$mm_2m_4m_5$</td>
<td></td>
</tr>
<tr>
<td>$mm_1m_2m_5m_6$</td>
<td></td>
</tr>
<tr>
<td>$mm_2m_4m_5m_6$</td>
<td></td>
</tr>
</tbody>
</table>

$m_{m_1m_2m_5}$, $m_{m_2m_4m_5}$, $m_{m_1m_2m_5m_6}$, $m_{m_2m_4m_5m_6}$, $m_{m_1m_2}$, $m_{m_2m_4}$ and $m_{m_2m_4m_5m_6}$. Table 2.1 shows the entries in the route cache of mobile node $m$ and Table 2.2 shows the corresponding entries in the information table. When the route $mm_1m_2$ is to be added to the route cache shown in Table 2.1, it is seen that this route is a subpath of the route $mm_1m_2m_4m_5$ already present in the route cache. So, this route is not added to the route cache. However it is added to the information table as shown as the element $x_6$ in Table 2.2. Similarly the elements $x_7$ and $x_8$ also correspond to identical routes or subpaths of already existing routes in the route cache.

### 2.3.2 Decision system

The decision system in a mobile node is used to predict the next hop for a particular destination. This next hop is called the *predicted next hop*. The decision attribute is taken as the next hop, and the predicted next hop is also known as the *predicted value of the decision attribute*.

The decision system in a mobile node $m$ is $\{U_m, A \cup \{d\}, V, \rho_m\}$, where $d$ is the decision attribute. Given a particular destination node,
Table 2.2: Information table with mobile nodes as attributes

<table>
<thead>
<tr>
<th></th>
<th>m</th>
<th>m₁</th>
<th>m₂</th>
<th>m₃</th>
<th>m₄</th>
<th>m₅</th>
<th>m₆</th>
</tr>
</thead>
<tbody>
<tr>
<td>x₁</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>x₂</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>x₃</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>x₄</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>x₅</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>x₆</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>x₇</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>x₈</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

the decision system is used to find the predicted next hop. Consider a particular destination $m_t \in M$. Let $d \in M$, the decision attribute, correspond to the next hop that should be taken by a packet from this mobile node $m$ to reach the destination $m_t$ in a route. So, $V_d = M$. Let $x \in U_m$ be the row corresponding to the route $m m_1 m_2 \ldots m_k$, $m_i \in M$, $i = 1, \ldots, k$, and $m_t$ is one of the $m_i$. It is noted that $\rho_m(x, d) = m_1$.

In this first study, a simple model is proposed. Let $B = \{m_t\}$, $B \subseteq A$. Consider the decision system $I_B = \{U_m, B \cup \{d\}, V, \rho_m\}$. $I_B$ is used in the following manner to determine the next hop that is to be taken by a packet to reach the destination $m_t$. For some $m \in V_d$ and $k \in V_{m_t}$, let $X_{m,k}$ denote the set $X_{m,k} = \{y \in U_m \mid (\rho_m(x, d) = m) \land (\rho_m(x, m_t) = k)\}$. In other words, the set $X_{m,k}$ consists of the elements in $U_m$ of the decision system $I_B$ that have $m$ as the next hop and the destination node $m_t$ has the value $k$ in the corresponding route (that is, the destination node $m_t$ is present or not present in the corresponding route according to $k$ being 1 or 0). Let $|X|$ as usual denote the count of elements in a set $X$. The value $m_n \in V_d$ is such that $|X_{m_n,1}| \geq |X_{m_i,1}| \forall m_i \in V_d, m_i \neq m_n$. In
other words, the value $m_n$ of the decision attribute $d$, is the node that is used as the next hop by the most number of entries that have this destination $m_r$ in the corresponding route.

This first model successfully applies information systems to mobile ad hoc networks by appropriately mapping the universe of elements, the attributes and their values. The information systems are maintained at each mobile node and decisions are taken based on the information available at each mobile node.

2.4 MOBILE AD HOC ROUTING USING THE DECISION SYSTEM

2.4.1 Routing based on the decision system (DSR$_+$)

A route is known to a mobile node $m \in M$ either because the route is in a packet that passes through this mobile node, or because this mobile node is in promiscuous mode and this route is in a packet that passes between two nodes that are within range of this mobile node. When a route is known to a mobile node it is added to the route cache only if it is not identical to a path or a sub-path of any other route already present in the route cache. However, every time a route is known to a mobile node, a row corresponding to this route is always added to its information table.

Let $m_s$ be the source node that wants to send a data packet to a destination node $m_t$. In the source node, as in DSR, a shortest route in the route cache if available, is placed as the source route in the data packet. If a route is not available in the route cache, a route discovery is done.
Then in the source node and in any intermediate forwarding node, 
the decision system explained in Section 2.3.2 is used to find the best next hop 
(Algorithm 1). When this next hop is found from the information table, only the 
rows that have the given destination are considered. If the given destination is 
a next hop, then the next hop itself is chosen as the destination. Else, the node 
that is used by most of the entries as next hop to reach the destination is chosen 
as the next hop.

**Algorithm 1** Finding a next hop in DSR+

```java
findNextHopDSRplus()
{
    foreach possible next hop nh do
        count[nh] = 0;
        foreach row in the information system do
            if the route corresponding to this row has nh as the next hop and will lead
to the destination then
                count[nh] = count[nh] + 1;
            end
        end
    end
    Find the next hop nh for which the value of count is the maximum and return;
}
```

If the next hop that is found from the information table is different 
from the one in the source route that is already in the data packet, this new next 
hop is appended to the source route in the data packet at the current node and a 
flag is set in the data packet to know that the source route taken from the source 
node has changed.

If a next hop cannot be determined from the information table or if 
the next hop found from the information table results in a loop, then if the flag 
in the data packet is not set earlier (source route from the source node has not 
changed), the data packet is forwarded according to the source route. Else, a 
route discovery is done (Algorithm 2).
Algorithm 2 Intermediate node choosing a next hop

```
interNodeChooseNextHop()
{
    Find a next hop ‘nh’ from the information system;
    if ‘nh’ will result in a loop then
        if flag not set in header then
            Route according to source route;
        end
    else
        Initiate route discovery;
    end
else
    Append ‘nh’ to the source route from the current node’s id;
    if ‘nh’ is not the same as the current next hop according to source route then
        Set flag in header of data packet;
    end
end
```

When an intermediate node is forwarding a data packet and it does not find the link to the next hop, it sends a route error with the information about the dead link to the source node. When the source node receives the route error, it deletes in each path of the cache the subpaths starting from the dead link. This deletion is also done in the information tables in the source node and the intermediate nodes that forwarded the route error.

2.4.2 Simulation environment

The network simulator ns2 is a discrete event simulator of the University of California at Berkeley and the VINT project (Fall and Varadhan 2002). ns2 is a widely used simulator for research in networking. It is an object oriented simulator, written in C++, with an OTcl interpreter as a front end. It provides
substantial support for simulation of TCP, routing and multicast protocols over wired and wireless (local and satellite) networks. It also provides support for implementation and testing of new protocols and applications.

The Monarch research group at Carnegie Mellon University extended the ns2 simulator to include wireless scenarios with mobile nodes. It allows simulation of wireless LANs and ad hoc networks. The ns2 simulator with Carnegie Mellon University’s extensions is used for the experiments and the results presented in this thesis.

The radio model in the simulator is based on the Lucent Technologies WaveLAN 802.11 providing a 2 Mbps transmission rate. The link layer modelled is the Distributed Coordination Function (DCF) of the IEEE 802.11 wireless LAN standard. This models the contention of nodes for the wireless medium. An implementation of Address Resolution Protocol (ARP) modelled after the BSD Unix implementation is used in the simulator to resolve IP addresses to link layer addresses.

Nodes in the simulation move according to the random waypoint mobility model. Each node begins the simulation by remaining stationary for pause time seconds. It then selects a random destination in a rectangular space and moves to that destination at a speed distributed uniformly between 0 and some maximum speed. The node stays in that position for pause time seconds before it chooses another destination. This process repeats throughout the simulation causing continuous changes in the topology of the underlying network. For the experiments in this thesis, constant bit rate (CBR) traffic sources are used. 512 byte data packets are used. A transmission range of 250 m is used.
For each set of parameters chosen for the simulation, multiple runs of the simulator are executed varying the initial positions of the mobile nodes, the positions to which the nodes move and the source-destination pairs that communicate at a time. An average of the values got in the multiple runs is then calculated to get the final results.

2.4.3 Performance metrics

The metrics used for performance evaluation in other MANET studies are:

(i) Packet delivery ratio: The ratio of the data packets delivered to the application layer of the destination to those sent by the application layer of the source node.

(ii) Average end-to-end delay: The average delay from when a packet is sent by the source node until it is received by the destination node.

(iii) Normalized control overhead: The ratio of the number of control packets sent to the number of data packets received in the application layer.

(iv) Average hop count: The average number of hops from the source node to the destination node.

Packet delivery ratio describes the loss rate that will be seen by transport protocols, which in turn effects the maximum throughput that the network can support (Broch et al. 1998).
2.4.4 Performance evaluation varying the number of connections

In this first study, performance is evaluated using packet delivery ratio and end-to-end delay. The simulation environment is as explained in Section 2.4.2. Initially, the mobile nodes are spread randomly over the network. Nodes move in a field with dimensions 670 m × 670 m with a maximum speed of 5 m/sec. The pause time is 2 seconds. The number of nodes is kept fixed as 15. The number of communicating source-destination pairs (connections) is varied from 5 to 12. Data packets are transmitted at a rate of 0.5 packets/sec. Simulations are run for 500 simulated seconds.

Each value in the tables and graphs shown in this subsection is the average of the values got in 10 simulation runs corresponding to 10 different scenarios. For each scenario, the communicating source-destination pairs and the initial positions of the nodes are changed randomly. The results are given in Table 2.3.

It is seen from Figure 2.1 that there is not much of difference in the packet delivery ratio of DSR and DSR+. In Figure 2.2, the variation of the average end-to-end delay for DSR and DSR+ with the increase in the number of connections is shown. It is seen that there is an improvement of 13% to 41% in the average end-to-end delay of DSR+, when the number of connections is less than 10. But when the number of connections is increased, the average end-to-end delay of DSR+ is not better than that of DSR.
Table 2.3: DSR vs. DSR⁺ varying the number of connections

<table>
<thead>
<tr>
<th>No. of Conn.</th>
<th>Pkt. Delivery Ratio (%)</th>
<th>Avg. Delay (secs)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DSR</td>
<td>DSR⁺</td>
</tr>
<tr>
<td>5</td>
<td>98.09</td>
<td>98.12</td>
</tr>
<tr>
<td>6</td>
<td>98.39</td>
<td>98.28</td>
</tr>
<tr>
<td>7</td>
<td>97.84</td>
<td>97.53</td>
</tr>
<tr>
<td>10</td>
<td>95.89</td>
<td>95.84</td>
</tr>
<tr>
<td>11</td>
<td>96.99</td>
<td>96.79</td>
</tr>
<tr>
<td>12</td>
<td>96.48</td>
<td>96.26</td>
</tr>
</tbody>
</table>

Since there is an improvement in the average end-to-end delay for less number of connections, the proposed method appears to have potential use and more performance evaluation is done. Since the number of connections played a role, the variation with the number of nodes is studied next. The number of connections is kept fixed at 10 and the number of nodes is varied from 15 to 50.
Figure 2.1: Comparison of packet delivery ratio of DSR and DSR\(_+\) for varying number of connections

Figure 2.2: Comparison of average end-to-end delay of DSR and DSR\(_+\) for varying number of connections
### Table 2.4: Comparison of DSR and DSR+ for varying number of nodes

<table>
<thead>
<tr>
<th>No. of Nodes</th>
<th>Pkt. delivery ratio (%)</th>
<th>Avg. delay (secs)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DSR</td>
<td>DSR+</td>
</tr>
<tr>
<td>15</td>
<td>69.01</td>
<td>69.19</td>
</tr>
<tr>
<td>20</td>
<td>78.9</td>
<td>77.72</td>
</tr>
<tr>
<td>25</td>
<td>92.19</td>
<td>88</td>
</tr>
<tr>
<td>30</td>
<td>92.43</td>
<td>89.48</td>
</tr>
<tr>
<td>35</td>
<td>94.12</td>
<td>91.71</td>
</tr>
<tr>
<td>40</td>
<td>95.77</td>
<td>92.94</td>
</tr>
<tr>
<td>45</td>
<td>95.77</td>
<td>93.97</td>
</tr>
<tr>
<td>50</td>
<td>96.46</td>
<td>92.31</td>
</tr>
</tbody>
</table>

#### 2.4.5 Performance evaluation varying the number of nodes

In this set of experiments, the number of communicating source-destination pairs is kept fixed at 10 and the number of nodes is varied from 15 to 50. The size of the region in which the nodes move is increased to 1000 m × 1000 m, the pause time to 20 seconds and the total simulation time is increased to 1000 simulated seconds. The other parameters are kept the same as that in the previous set of experiments. Nodes move with a maximum speed of 5 m/sec and data packets are transmitted at a rate of 0.5 packets/sec. The simulation environment setup is as explained in Section 2.4.2. The performance is evaluated using the metrics, packet delivery ratio and average end-to-end delay. The results are given in Table 2.4.

It is seen from Figure 2.3 that there is not much of difference in the packet delivery ratio of DSR and DSR+ even when the number of nodes is increased. The average end-to-end delay of DSR+ is seen to be better (an average of 20%) as the number of nodes increases (Figure 2.4). So, in a larger region, with more number of nodes, the average end-to-end delay of DSR+ is
seen to be better.

When there are more number of nodes in the network, the network is dense and here, the nodes move in a moderate speed. The number of possible routes between any two nodes will be more. So, the number of routes learnt is more. For example, route requests will learn more routes than when there are less number of nodes. This contributes to better learning and the next hop chosen is such that the delay is reduced. The packet delivery ratio is high in DSR itself for these scenarios. So there is not much scope for further improvement in the packet delivery ratio.

![Comparison of packet delivery ratio of DSR and DSR+ for varying number of nodes](image)

**Figure 2.3:** Comparison of packet delivery ratio of DSR and DSR+, for varying number of nodes
Figure 2.4: Comparison of average end-to-end delay of DSR and DSR⁺ for varying number of nodes

2.5 SUMMARY

The use of the information system and decision system of Rough Set Theory in mobile ad hoc routing was introduced in this chapter. In the proposed routing protocol DSR⁺, the information system was incorporated into the mobile node, and the corresponding decision system was used in choosing a next hop for routing data. It was seen that DSR⁺ performed better than DSR when the number of connections was less.