Agent Reliability Estimate

4.1 Introduction

Mobile agents can be very useful over message passing systems in wireless applications as they could help address the constraints of the underlying environment. For example, as mentioned before, mobile agents can move to the place where information is stored and process data locally, discarding data that is not relevant and therefore need not be communicated through the wireless network. As another example, they can transport themselves and their data through networks, by continuously evaluating alternatives based on information about their environment. Thus agents can be provided with a list of nodes to visit or they can choose their trail dynamically based on information they extract from the environment. When the environment is dynamic as in Mobile Ad hoc NETwork (MANET), it itself affects the performance of the mobile agents.

In this chapter, MAS is defined to be a system consisting of a number of different groups of agents where each group accomplishes an independent task. These agents are assigned some task by their owner for which they need to migrate from one mobile node to another in MANET. In MANET, like in any other mobile computing system, mobile nodes access information through wireless data communication at any time and everywhere (motion and location independence) [1]. The MANET environment is inherently uncertain (as nodes can only have partial information about its neighbourhood) and dynamic because of node mobility. Moreover performance of the network depends on how well the mobile nodes forward each others’ traffic. These features affect both availability and reliability of the services of the system. Hence reliability of underlying network becomes a factor that may affect the performance, availability and strategy of mobile agent systems [25], [52].

However the scalability/reliability issue of MAS has already been highlighted in [94], although the work does not focus on MANET.

The present system (S) consists of M number of independent agents that may move in the underlying network. The reliability of S is defined as the probability that S is operational during a period of time [106]. Consequently S is said to be fully operational if all M mobile agents are functional, that is, all have completed their task and there is no software fault [77]. On the contrary S is fully down if all M mobile agents are
fully non-operational. Moreover, S is said to be partially operational if some of the M mobile agents are operational. For applications like service discovery, network clustering, partially operational agents can be used as failure of some agents do not stop the whole application. But in applications like e-commerce, if the server node crashes the whole application comes to a halt. So for such applications partially operational system is of no use. In this chapter those distributed applications are considered where partially operational MAS is acceptable.

4.2 Modeling MANET

The underlying network is modeled as an undirected graph $G = (V, E)$ where $V$ is the set of mobile nodes and $E$ is the set of edges among them. Let the network consist of $N$ nodes, thus, $|V| = N$ that may or may not be connected via bidirectional links $(e)$. The following assumptions are made ([106] [107]):

1. The network graph has no parallel (or redundant) links or nodes.
2. The network graph has bi-directional links.
3. There are no self-loops or edges of the type $(v_j, v_j)$.
4. The states of vertices and links are mutually statistically independent and can only take one of the two states: working or failed.

With these assumptions MANET is modeled in the following two ways.

4.2.1 Non Homogeneous Poisson Distribution

Depending on a given probability a link may either exist or cease to exist at a particular point in time. In MANET there can be several reasons for link failures including relative mobility of the nodes, obstacle in the signal path causing shadowing or diffraction of the signal [104], frequency selective fading, heavy rainfall absorbing significant signal power, selfish behaviour of the forwarder node etc. As is evident these factors are transient and most of the time independent in nature. Thus the average rate of link failure is not constant over time. So, the event of link failure can be modeled using a Non Homogeneous Poisson Process (NHPP) [108]. In NHPP the fixed rate parameter of Poisson distribution becomes a function of time. So the mean rate of node movement itself varies with time. In NHPP the number of events in disjoint intervals are independent random variables. Since link failures at different times of the day are independent of one another, NHPP seems to be a suitable choice to depict such a scenario. However
another scenario may be where all nodes may move and hence links may fail with equal probability.

4.2.2 Smooth Random Mobility Model

The movement pattern of users plays an important role in performance analysis of algorithms and protocols for mobile ad hoc networks [109]. Link failure rates, performance of routing protocols and hence strategy of agent migration depend on relative mobility of the nodes. Whereas in cellular networks there exists a number of approaches that model the macroscopic movement behavior of users (how much traffic moves from one cell to the other), but in MANET “microscopic” model is required as we need to focus on mobility of individual nodes. The choice of mobility model has a significant effect on obtained results (this is shown in later chapters). If the model is unrealistic, it may lead to invalid conclusions [109].

In this work Smooth Random Mobility model (SRMM) is used to simulate node mobility. It is a random mobility model on a microscopic scale [109]. Here movement of one node bears no similarity with others. But speed at a time instant depends on the speed at earlier time instant and a node can only change direction incrementally at certain time instants, for example, when speed is zero. Thus it prevents the nodes from taking sharp turns or making sudden stops to ensure smooth movement. This model is simple, not biased for any particular application scenario (both cars and pedestrians carrying mobile devices can be modelled) and yet not unrealistic like random way point mobility model [36].

SRMM uses two stochastic processes:

- one process determines at what time a mobile node changes its speed and
- the other process determines when the direction is changed [100]

In SRMM [100] Poisson event is utilized to determine the time of change in speed of a node. A new target speed is chosen from $[0, V_{\text{max}}]$ where 0 and $V_{\text{max}}$ are given higher preference and rest of the values are equally likely. Once a target speed is chosen the current speed is gradually changed to attain the new target speed according to the present acceleration $a(t)$. $a(t)$ is chosen from $[0, a_{\text{max}}]$ where each values are equally likely. Here $V_{\text{max}}$ and $a_{\text{max}}$ denote the maximum speed and maximum acceleration of each user respectively and may be different for different users. For example, for vehicular traffic, these will have higher values than pedestrians. Thus, speed of the $i^{\text{th}}$ user at time instant $t$ depends on its speed at the previous time instant $v_i(t - \Delta t)$ and its present acceleration $a_i(t)$ as in [100],
4. Agent Reliability Estimate

\[ v_i(t) = v_i(t - \Delta t) + \Delta t \times a_i(t) \]  \hspace{1cm} (4.1)

A new target direction is chosen only when \( v_i(t) = 0 \). The stop turn and go [100] behavior is simulated here. The target direction is uniformly distributed between \([-\pi/2, \pi/2]\) with \(-\pi/2\) and \(\pi/2\) having higher priorities [100]. At every time instant direction \((\Delta \phi_i(t))\) changes incrementally \((\Delta \phi_i(t))\) unless it attains the target direction. Here \((\Delta \phi_i(t))\) represents the maximum direction change that node \(i\) can make during time \(\Delta t\). Thus, as in [100],

\[ \phi_i(t) = \phi_i(t - \Delta t) + \Delta \phi_i(t) \]  \hspace{1cm} (4.2)

Now, using this speed at previous time instant, acceleration, and direction, we can estimate the position \((x_i, y_i)\) of the node \(i\) at \((t + \Delta t)\) as

\[ x_i(t + \Delta t) = x_i(t) + \Delta t \times v_i(t) \times \cos \phi_i(t) + 0.5 \times a_i(t) \times \cos \phi_i(t) \times \Delta t^2 \]  \hspace{1cm} (4.3)

\[ y_i(t + \Delta t) = y_i(t) + \Delta t \times v_i(t) \times \sin \phi_i(t) + 0.5 \times a_i(t) \times \sin \phi_i(t) \times \Delta t^2 \]  \hspace{1cm} (4.4)

The distance between a pair of nodes \((d_{s_{ij}})\) can be calculated as follows

\[ d_{s_{ij}} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \]  \hspace{1cm} (4.5)

The probability of link existence \((P_{\text{link}})\) not only depends on the distance between the nodes but is also very much dependent on environmental factors. So, even when two nodes remain within the transmission range of each other, due to factors like signal fading, shadowing, diffraction etc., quality of transmission can degrade appreciably [110]. The average received power \((p_r)\) is a function of the distance between the transmitter and the receiver. Here we take the two-ray model for radio propagation in order to show how the transmitted signal power \((p_t)\) suffers from multipath propagation while reaching the receiving end. Thus, \(p_r(d)\) can be stated as mentioned below [103]

\[ p_r(d) = p_t G_t G_r \frac{h_t^2 h_r^2}{d^4} \]  \hspace{1cm} (4.6)

Where \(h_t, h_r\) denote transmitter and receiver antenna heights and \(G_t\) and \(G_r\) denote transmitter and receiver antenna gains respectively. In free space, the received power varies inversely to the square of the distance but here the exponent is assumed to be 4 to indicate the presence of a medium.

Here connectivity between the nodes is formed by calculating received power. When
4. Agent Reliability Estimate

exceeds a certain threshold, the corresponding link is assumed to exist.

Breadth First Search (BFS) technique [28] is applied iteratively to the connectivity graph (thus formed using NHPP or SRMM) unless all nodes are given a cluster id. If there is a path between two nodes in MANET then both nodes share the same cluster id.

4.3 Modeling Mobile Agents on MANET

Mobile agents can be modelled as tokens traveling over the network from one node to another (if the nodes are connected and hence share the same cluster id) based on some strategy as needed by the underlying applications to accomplish its task [78].

An agent starts its journey from a given owner and moves from one node to another at its will. A dynamic planning strategy can be adopted where each agent is expected to visit \(\leq\) number of nodes in the network) nodes in the network to accomplish its task [102]. The agent at a node at any point can randomly choose any other (preferably neighboring) node to be its next destination.

Alternatively the owner can provide a priority list to the agent, which contains a list of node ids that are most beneficial migration sites (for the application that deployed that particular agent) [111]. So, an agent will always try to visit those nodes from the priority list as its first preference.

But this movement is successful only if the nodes are connected. We assume that cases of collisions (if any) are taken care of by the underlying MAC protocol. In practice the routing algorithm will determine the path that an agent may take to reach a certain destination. Here this scenario is abstracted as, if there is a path the agent will succeed otherwise the agent will fail to migrate. But the transient characteristics of the environment cannot be handled by the MAC or routing protocol either. So, we associate a probability with the movement to consider the influence of these factors. For example, the routing table may not be updated properly or the link quality may have degraded so much (due to increased noise level) that the agents are unable to migrate. Thus, if an agent residing at \(\text{MN}_A\) decides to move to \(\text{MN}_B\) (connected to \(\text{MN}_A\)) then the agent successfully moves to \(\text{MN}_B\) with probability \(p_{tr}\). Here \(p_{tr}\) reflects the unpredictable background noise level as mentioned above. For instance, noise level may increase due to heavy rainfall. Let us suppose that at an instance \(t\), the MANET consists of five nodes namely \(\text{MN}_A\), \(\text{MN}_B\), \(\text{MN}_C\), \(\text{MN}_D\) and \(\text{MN}_E\) and their connectivity is as shown in Figure 4.1(a). The dotted line represents an erroneous link. It is assumed that all nodes provide appropriate host platform for the agents and the agents may update their migration pol-
icy on the fly depending on the information they extracted from the visited nodes. An agent x (say) residing at node MN_A therefore, chooses its next destination MN_B almost randomly giving more preference to the nodes in the priority list. If that destination is not visited before and if there is a path then x moves to its new location with probability $p_{tr}$. However if network topology changes (Figure 4.1(b)), MN_B becomes an isolated node and thus unreachable. It may also be such that the capacity of the link (from MN_A to MN_B) is lower than that needed by x. So as far as routing algorithm is concerned, a link between MN_A and MN_B exists, but for agent x, the capacity of the link is not sufficient. So MN_B is unreachable for x. So x will not be able to move to MN_B. In the next time instant x may retry or try to choose its next destination depending on its migration policy.

Thus the agents themselves try to overcome the transient faults. In this work system

![Figure 4.1: An example of MANET configuration: (a)At instant t (left); (b)At t+Δt(right)](image)

performance is measured in terms of MAS reliability.

### 4.4 Reliability Estimation of MAS

The reliability of MAS is measured with respect to the network status and its conditions (for example, connectivity of the links, path loss probability etc.). As stated earlier each agent is expected to visit N ($\leq$number of nodes in the network) nodes in the network to accomplish its task. Each agent starts its journey from a given node, which acts as its owner.

Since performance of the mobile agents depend on MANET conditions, reliability of
4. Agent Reliability Estimate

the system \( R_s \) can be defined as

\[
R_s = R_{\text{MAS}} \mid R_{\text{MANET}}
\]  \hspace{1cm} (4.7)

where system reliability is shown as a conditional probability of the reliability of MAS \( R_{\text{MAS}} \) given reliability of MANET \( R_{\text{MANET}} \). \( R_{\text{MANET}} \) can be treated as an accumulative factor of \((1-P_{\text{Node}})\) and \( P_{\text{Link}} \). The failure probability of the nodes, \( P_{\text{Node}} \) can be represented by some probability distribution. \( P_{\text{Link}} \) can be treated as a combination of the mobility model and the probability that \( p_r \) is at an acceptable level. Here \( p_r \) denotes the received power at node \( j \) after traversing distance \( d_{ij} \) from sender node \( i \).

The term reliability indicates the probability of failure-free operation for a specified period of time in a specified environment. Thus reliability can only be predicted for a certain time period. Here individual agent reliability \( \lambda_i(t) \) on the underlying MANET is calculated as follows:

If an agent \( i \) can successfully visit \( n \) nodes out of \( N \) (desired) by time \( t \) then it has accomplished \( n/N \) portion of its task. Thus, reliability in this case will be as follows

\[
\lambda_i(t) = \frac{n}{N}
\]  \hspace{1cm} (4.8)

But if the application requires all \( N \) nodes to be visited in order to fully accomplish the task and in all other cases the task will not be considered to be done, reliability will be:

If an agent can successfully visit all \( N \) nodes desired (by time \( t \)) then it has accomplished its task. Thus, reliability in this case will be 1. In all other cases it will be 0.

Above definitions of agent reliability works only if there is no software failure of the agent. Now, the probability that the MAS is operational i.e., reliability of MAS \( R_{\text{MAS}} \) can be calculated as the mean of reliability of all its components, that is, the agents in this system.

\[
R_{\text{MAS}} = \frac{\sum \text{Agent Reliabilities}}{M}
\]  \hspace{1cm} (4.9)

Finally to calculate \( R_s \) in equation 4.7 two algorithms are proposed in the following section.

4.5 Algorithms for Estimation of Reliability

The first algorithm for reliability estimation assumes the following:
4. Agent Reliability Estimate

1. NHPP is used to simulate link conditions and

2. Each agent is asked to visit N nodes in the network.

The algorithm BasicAgentReliabilityEstimation() is summarized in Algorithm 4.

Algorithm 4: BasicAgentReliabilityEstimation(). An algorithm for estimating MAS reliability on MANET $G = (V, E)$.

**Input:** $G = (V, E)$

**Output:** $R_s$

begin

1. Repeat Q times do

2. Initialize n (the number of mobile nodes successfully visited by an agent) to 0 and a source for the mobile agent.

3. Input network configuration $(V, E)$ in the form of an edge list.

4. for simulation time T

5. Network topology is predicted and list of connected components is calculated using nhppMANET() listed in Algorithm 5.

6. for all agents (M) in the system

7. Individual software reliability of the agents $r_i(t)$ is calculated according to Weibull distribution.

8. while there is a connected but yet unvisited destination for an agent do

9. The agents perform their job on this modified graph.

10. From the starting position an agent randomly choose a yet unvisited node to be its next destination.

11. if that destination falls in the same cluster as it is now residing then

12. Increment n by 1.

13. Calculate agent reliability following equation 4.8.

14. Reset the value of n.

15. Calculate $\lambda(t) = \frac{1}{M} \sum_{i=1}^{M} \lambda_i(t)r_i(t)$

16. Calculate system reliability as $R_s = \frac{1}{Q} \sum_{q=1}^{Q} \frac{1}{T} \sum T \lambda(t)$

end

Another algorithm for reliability estimation AgentReliabilityEstimation() is summarized in Algorithm 6 where the following are considered

1. SRMM is used to simulate the effect of node mobility.

2. The probability of the existence of a link is calculated according to equation 4.6 to cover multipath propagation effect of radio signals.
4. Agent Reliability Estimate

Algorithm 5: nhppMANET(). An algorithm for generating MANET $G = (V, E)$ configuration at a time instant using NHPP.

Input: Initial list of edges $E$
Output: List of connected components

begin
1. Create $E' \subseteq V \times V$ using NHPP (or Uniform) distribution. Likewise a probability $p_i$ is associated with each link $e_i$ that is calculated using NHPP.
2. if this falls within the range $[p,1]$ for $0 < p < 1$ then
3. $e_i$ is assumed to be operating.
4. else
5. It is deleted from $E'$ as $e_i$ is failed.
6. Software/hardware failure of the nodes is modeled using another NHPP distribution.
   // Node failure can be simulated by deleting e's from $E'$ further that are incident on the failed node $v \in V$.
7. Use BFS unless all connected subgraphs are assigned a proper cluster id.
   // Thus an isolated node is also a cluster.
end

3. A mobile agent prefers to select a destination, which is not visited before, from the priority list. If it finds a route (that is if the source and destination share the same cluster id) then it moves with a certain probability and this continues. Otherwise in the next time instant the agent chooses another unvisited node from its priority list for migration.

It is to be mentioned that BasicAgentReliabilityEstimation() shown in Algorithm 4 calls nhppMANET() (listed as Algorithm 5) and AgentReliabilityEstimation() shown in Algorithm 6 calls srmMNANET() (listed as Algorithm 7) for every move of the mobile agent. Since in a typical adhoc scenario the nodes are also not static during the entire tour of the mobile agents, so mobility of nodes is considered and network topology changes. With time some nodes may go far apart from each other causing links to be deleted as well as a few others may come closer creating links between them.

If an agent fails to move because of background noise level, then it may retry or try out other alternatives according to steps 13-14 of AgentReliabilityEstimation() listed as Algorithm 6.

Here it is assumed that in order to accomplish a task the agents need to visit all nodes in the network. But this parameter can be changed and both algorithms will still
4. Agent Reliability Estimate

**Algorithm 6**: AgentReliabilityEstimation(). An algorithm for estimating MAS reliability on MANET $G = (V, E)$.

**Input**: Initial position, initial speed and maximum acceleration of each node  
**Output**: $R_s$

```plaintext
begin

1 Repeat Q times do
2 Initialize n to 0 and a source for each mobile agent.
3 for simulation time T
4   Simulate mobility and find connected components using `srmmMANET()` in Algorithm 7.
5   for all agents (M) in the system
6     Individual software reliability of the agents $r_i(t)$ is calculated according to Weibull distribution.
7     while there is a connected but yet unvisited destination
8       for an agent do
9         An agent will prefer to choose an unvisited node from the priority list to be its next destination.
10        / All unvisited nodes are equally likely
11        if that destination falls in the same cluster [102]
12          The agent moves there with probability $p$, the instantaneous background noise in the network.
13        if it succeeds then
14          n is incremented by 1.
15        if migration fails despite several attempts then
16          It tries out other alternatives, needed by the application.
17          Calculate agent reliability following equation 4.8.
18          Reset the value of n.
19          Calculate $\lambda(t)$ as in step 15 of Algorithm 4
20          Calculate system reliability as $R_s$ as in step 16 of Algorithm 4

end
```

72
4. Agent Reliability Estimate

Algorithm 7: srmmMANET(). An algorithm for configuring MANET $G = (V, E)$ at a time instant.

Input: Initial position, initial speed and maximum acceleration of each node
Output: List of connected clusters

begin
1. To simulate the effect of node mobility create $E' \subseteq V \times V$ using SRMM as follows.
2. The $v_i(t)$ and $\phi_i(t)$ are calculated using equation 4.1 and 4.2.
3. The position of each MN is updated for the next time increment following equations 4.3 and 4.4.
4. Distance between each node pair is calculated using equation 4.5.
5. if $P_r$ calculated following equation 4.6 is $> \text{a threshold}$ then $E'$ is populated.
6. Node failure probability is also calculated as in step 6 of nhppMANET() (Algorithm 5).
7. BFS is used to find the connectivity clusters as in step 8 of nhppMANET() (Algorithm 5).

end

work if lesser number of nodes is needed to be visited. Here a basic assumption is that the agent can always retract back to its owner.

It may be seen in practice that in a network some nodes have rich information and the agents tend to move to those nodes as their next destination over the other. That is why the nodes are prioritized and agents are provided a priority list rather than randomly selecting the next destination in AgentReliabilityEstimation() (Algorithm 6). Here the priority list from owners is given to the agents but the agents may also learn about such rich nodes from their experience and share this information with others.

4.6 Simulation for Reliability Estimation

The simulation is carried out in Java and it can run in any platform. First the results of BasicAgentReliabilityEstimation() (Algorithm 4) is presented [102]. It is followed by the results of AgentReliabilityEstimation() (Algorithm 6) [111] in the following subsections.

4.6.1 Using Non Homogeneous Poisson Distribution

An initial configuration of MANET would be assumed. Afterwards due to mobility few links may fail and still a few may be revived also according to NHPP considering the transient nature of the faults. Thus whenever link existence probability (calculated
4. Agent Reliability Estimate

According to NHPP) falls below a certain threshold, the link is considered to be failed. Thus connection matrix is formed at a particular time instant.

![Image](https://via.placeholder.com/150)

Figure 4.2: (a) Complete graph of 6 nodes; (b) Generated graph after applying the algorithm

Let us take an example scenario of 6 mobile nodes and apply BasicAgentReliabilityEstimation() (Algorithm 4). The experiment is conducted with 2 mobile agents being spawned from MN0 and MN1 respectively. The experiment is done with a complete network graph having N=6 nodes (Figure 4.2(a)). So all the host platforms (nodes) have a direct link via which a mobile agent may move to another host platform. After applying step 5 of BasicAgentReliabilityEstimation() (Algorithm 4) the resulting network configuration is as shown in Figure 4.2(b). Since the graph is already connected all nodes will come under a single cluster after applying BFS. So unless and otherwise there is a software fault (simulated according to Weibull distribution following step 7 of BasicAgentReliabilityEstimation() listed as Algorithm 4) agents will continue to move to any host platform.

![Image](https://via.placeholder.com/150)

Figure 4.3: Variation of reliability with varying network connectivity

Thus the overall reliability for this particular configuration becomes 1 irrespective of
4. Agent Reliability Estimate

Table 4.1: Data for Reliability Calculation

<table>
<thead>
<tr>
<th>Number of Edges</th>
<th>Reliability</th>
<th>Initial Connectivity</th>
<th>Graph Connectivity</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>1</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>14</td>
<td>0.833</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>13</td>
<td>0.583</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>12</td>
<td>0.416</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>11</td>
<td>0.416</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>10</td>
<td>0.333</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>9</td>
<td>0.333</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>8</td>
<td>0.333</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>7</td>
<td>0.333</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>6</td>
<td>0.25</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>5</td>
<td>0.167</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>4</td>
<td>0.167</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>3</td>
<td>0.167</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>2</td>
<td>0.167</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>1</td>
<td>0.167</td>
<td>no</td>
<td>no</td>
</tr>
</tbody>
</table>

the number of agents if all of them are active. Reliability is expected to drop when the resulting graph is not connected. To study this, edges are removed gradually and its effect on reliability is observed in Table 4.1. It is evident from the table and corresponding Figure 4.3 that reliability of MAS drops appreciably when the resulting graph loses connectivity and becomes divided into components. Hence it may be concluded that higher the mobility in the network lesser will be the reliability of the system. However the system can perform quite reliably even with higher node mobility if there is no local interference and the nodes move in groups (group of students in a campus for example). This is also proved by the experimental results of AgentReliabilityEstimation() (Algorithm 6).

4.6.2 Using Smooth Random Mobility Model

The initial positions of the MNs are given along with their initial speed and the maximum acceleration that can be attained by them. These initial values are needed by SRMM. The movement of the nodes is assumed to be bounded within a specified simulation area as in [100]. But the nodes are allowed to move anywhere within the simulation plane. All agents of the same group start from the same node, the owner. We have taken an in-
4. Agent Reliability Estimate

Table 4.2: Priority List of the Agents

<table>
<thead>
<tr>
<th>Agent ID</th>
<th>Priority List</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent 1</td>
<td>MN₂, MN₄</td>
</tr>
<tr>
<td>Agent 2</td>
<td>MN₁, MN₃</td>
</tr>
<tr>
<td>Agent 3</td>
<td>MN₄</td>
</tr>
<tr>
<td>Agent 4</td>
<td>MN₁, MN₂</td>
</tr>
</tbody>
</table>

Figure 4.4: Smooth movement of the nodes

stance where there are ten nodes in the network. Four mobile agents are deployed by four different owners and they start their journey from their respective owners. Thus agents 1, 2, 3 and 4 start their journey from nodes MN₁, MN₂, MN₃ and MN₄ respectively and roam around the network to accomplish respective task. The positions of the different nodes for 30 seconds (duration of this simulation experiment) are shown in Figure 4.4. The smooth movement of the nodes is obvious from the figure itself. Behaviour of the agents for the first four successive time instants is given in Figure 4.5 and is detailed below. Thus, an application (for example, service discovery) running on MN₁ deploys agent 1. The experiment finds the number of nodes that are successfully visited by these agents, which indicates the progress of its task (how many services the agents discover for a MANET) and consequently the reliability of the agents will be calculated. Average reliability of all the agents taken over a certain time period for a number of simulations represents the reliability of the MAS despite the uncertainties of MANET. So, all nodes have same priority in this MANET. However, the adopted migration policy gives some nodes more weightage over the others (step 8 in AgentReliabilityEstimation() listed in Algorithm 6) indicating the fact that all destinations are not equally likely. The agents are given a priority list by their respective owners as shown in Table 4.2. For example,
visiting nodes MN_2 and MN_4 will be most beneficial for agent 1 and so on.

Every 3 seconds the positions of the nodes are updated according to SRMM (shown in Figure 4.4). Connectivity among the nodes is calculated according to the Two-ray model following equation 4.6. As shown in Figure 4.5(a), MN_9 is isolated from the MANET initially. But eventually it finds MN_10 within its range and hence can connect itself to the network (Figures 4.5 (b), (c) and (d)). The network topology at 4 successive time instants is shown in Figure 4.5(a), (b), (c) and (d). Agents are also shown in Figure 4.5 by callouts along with a numeral to indicate agent ids. The dotted ones (callouts) represent the starting position and the bold ones (callouts) represent end point of their journey at that time instant.

Figure 4.5(a) indicates a disconnected network graph for the MANET with three clusters consisting of

1. MN_9 (single node cluster)
2. MN_3 and MN_8 (two node cluster)
3. MN_1, MN_2, MN_4, MN_5, MN_6, MN_7 and MN_10 (seven node cluster)

Here any agent can move to any destination it wants to, within its cluster. The agents start their journey in such a scenario.

While the nodes move and form a network configuration as shown in Figure 4.5(b), the agents also start migrating in the network. The network connectivity changes as MN_9 now comes within the transmission range of MN_10 and hence becomes connected to one of the clusters. So the MANET now contains two clusters, (one containing MN_3 and MN_8 and the other containing the rest). Since agent 1 gives priority to MN_2 and MN_4 over the others, so agent 1 first visits MN_4. For similar reasons, agent 2 visits MN_1 (from MN_2) and agent 4 visits MN_2 (from MN_4) respectively. But agent 3 cannot migrate successfully as node MN_4, the highly beneficial migration site for agent 3, lies in a cluster different from where agent 3 is.

Network connectivity changes a little in the next 3 seconds as indicated in Figure 4.5(c). So, agents 1 and 4 make successful migrations to their preferred destinations such as MN_2 (from MN_4) and MN_1 (from MN_2) respectively. However, MN_3, a highly beneficial migration site for agent 2 falls in a different cluster than MN_1 (where agent 2 currently resides). Consequently, agent 2 cannot make any migration but stays at MN_1. Moreover due to transient characteristics, the link between nodes MN_3 and MN_8 becomes erroneous. As a result agent 3 makes an unsuccessful attempt (step 10 in the Algorithm 6) to migrate to MN_8 (from MN_3) and stays at MN_3. As the agents can
Figure 4.5: Network graph and the position of the agents (a) At time instant \( t = t_0 \); (b) At time instant \( t = t_0 + \Delta t \); (c) At time instant \( t = t_0 + 2\Delta t \); (d) At time instant \( t = t_0 + 3\Delta t \)
migrate with a given probability, even if nodes fall in the same cluster, an agent may not be able to make a successful migration. This scenario indicates the notable effect of transient errors on the performance of MAS.

Finally in the next 3 seconds the collection of nodes form a connected graph as MN\(_3\) comes within the transmission range of MN\(_1\). Now the agents can migrate to any other node with a certain probability (step 9 of Algorithm 6). Thus, agents 1 and 4 migrate to MN\(_6\) (from MN\(_2\)) and MN\(_5\) (from MN\(_1\)) respectively. Agents 2 and 3 finally find their most beneficial migration sites (MN\(_3\) for agent 2 and MN\(_4\) for agent 3) reachable and attempt to make successful migration.

In this way, the simulation is continued and the nodes in the MANET continue to form different network configurations affecting agent migration. The value for received power is taken to be 16dBm. The other parameters like antenna gains is considered to be 2.2dBi, the height is taken to be 2m and the transmitting power is taken to be 20dBm [104].

At the end of the 30\(^{th}\) second, agents 1, 2 and 4 finish migrating to 9 nodes each (including their owners) out of all 10 nodes in the MANET accomplishing \(\frac{9}{10}\), that is, 90% of their tasks each. However agent 3 was able to cover only 7 out of 10 nodes (70%) because the agent was stuck in the two node cluster (consisting of MN\(_3\) and MN\(_8\)) for quite some time (shown in Figure 4.5(a), (b) and (c)), thus, accomplishing only 70% of its task. This scenario shows the effect of MANET configuration on the performance of MAS. Thus, the overall reliability of MAS comes out to be \((3 \times 0.9 + 0.7)/4 = 0.85\), that is, 85%. If another simulation run is carried out for the same amount of time, then the overall reliability comes out to be 0.825. If we use Monte Carlo simulation for a number of times (Q=100 onwards) the overall reliability tends to converge to 0.53 (as shown in Table 4.3). Thus, with a MANET of 10 nodes moving according to SRMM, the MAS where the agents almost randomly choose their neighbor and migrate, will be 53% reliable.

Table 4.3: Variation of Reliability with No. of Monte Carlo Simulation Steps

<table>
<thead>
<tr>
<th>Q</th>
<th>Reliability</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.539</td>
</tr>
<tr>
<td>100</td>
<td>0.5315</td>
</tr>
<tr>
<td>500</td>
<td>0.5319</td>
</tr>
<tr>
<td>1000</td>
<td>0.5314</td>
</tr>
<tr>
<td>2000</td>
<td>0.5312</td>
</tr>
<tr>
<td>10000</td>
<td>0.5319</td>
</tr>
</tbody>
</table>
4. Agent Reliability Estimate

For rest of the experiments, the default values of different parameters are listed below:

1. Simulation time is taken to be 1 hour;
2. N is taken to be 40 unless stated otherwise;
3. Parameters like received power, antenna gains are kept same as mentioned in the above example;
4. Unless otherwise stated M is taken to be 30;
5. The number of agents groups is taken to be 4;

![Variation of Reliability with No. of Nodes](a)
![Reliability Variation with Differing Acceleration](b)

Figure 4.6: (a) Variation of reliability with increasing network size; (b) Variation of reliability with greater variation of node accelerations

With four (4) groups and a total of 30 mobile agents, if size of MANET grows, reliability is found to drop eventually as shown in Figure 4.6(a). Since the network is already divided into a number of components, addition of new nodes may increase size of some components or form a new component itself. Thus the agents stuck in one component cannot move to the other. Consequently agents could not migrate to the nodes residing in other components. So the ratio of successful agent migration reduces as more nodes become unreachable for an agent. This results in the gradual fall in reliability with increasing N.

Now let us look into the matter in more detail for MANET with fast moving nodes. The maximum acceleration of the nodes is varied to yield different standard deviation for a given mean acceleration. It is to be noted that these acceleration values indicate the maximum acceleration that a node can achieve. Not necessarily all nodes use their maximum limit always. Rather it depends on the difference between the current and target speed of the nodes (following SRMM in [100]). When the average of all the
maximum acceleration that a node can attain is 0.75, we obtain the reliability value for standard deviation = 0.1, 0.2, 0.3, 0.4 and 1. Similar have been done when average of maximum acceleration is kept at 1.5 and 3 as shown in Figure 4.6(b). It is apparent from the result that when all nodes have the same variance in speed, if the overall MANET nodes are slower then obviously, the nodes will remain closer to each other implying higher reliability. In most cases for a given standard deviation, higher mean implies lower reliability. On the contrary, for a given mean, higher the standard deviation, lesser will be the reliability. This indicates that when all nodes move with comparable speed (lower standard deviation), for example, group movement in disaster relief or military operations, overall reliability improves. But when some nodes lag behind others, reliability of MAS would get hampered as the MANET breaks into a number of clusters.

![Reliability Variation with Differing Acceleration](image1)

![Variation of Reliability with Noise](image2)

Figure 4.7: (a) Variation of reliability with faster nodes; (b) Reliability variation with noise

The experimental observations are valid irrespective of the MANET nodes being slower or faster. Thus, in Figure 4.7(a), the points at the peak of the curve yield lower standard deviation. But if the mean goes even higher, that is for faster MANET, the reliability of MAS reduces as shown in Figure 4.7(a).

The effect of background noise is observed in Figure 4.7(b). For instance, in crowded regions, many mobile nodes are communicating simultaneously. Since all nodes use the same open shared broadcast medium, radio signal transmitted by one node interferes with others. So the receiver would not be able to decode the signal if the received signal power is low. Thus, a weak signal having signal power of 8dBm could not be decoded in crowded areas. But for environment with lower interference, such as highways or countryside, the transmission range increases, enabling weaker signals (having power of 8-15dBm) to be detected and decoded properly. Hence network connectivity improves making MAS more reliable. But when the interference is so high that the required signal strength is 15dBm higher (from 8dBm to 23dBm) system reliability degrades only by 15%
4. Agent Reliability Estimate

(from 0.55 to 0.4 approximately). This indicates that agents can tolerate background interference and still perform fairly well.

It can be observed that if node movements are allowed only at the beginning before the mobile agents start their task, then performance of the algorithm does not vary appreciably with number of mobile agents deployed in the system. But if the situation is made more realistic by allowing node movements in between agent migration, then reliability of MAS varies with its size as shown in Figure 4.8(a). As long as the number of agent groups remains fixed, the increasing size of MAS (in terms of M) does not seem to affect reliability greatly. But if the number of agent groups increase, even for a fixed number of agents, reliability improves and slowly reaches a stable state. This result is significant as it shows that a large number of applications deploying different types of agents (having different migration pattern) does not hamper the reliability of MAS. Rather they are able to cover different parts of the network and better exploit the denser portion of MANET. So, an increasing number of agent groups yield better performance than a single group of agents of comparable size. This is because the agents from the same group have similar migration pattern, they start from the same region of MANET and tend to face similar connectivity problems.

![Figure 4.8: (a) Reliability variation with increasing M and agent groups G; (b) Effect of varying priority list size on MAS reliability](image)

Let us now concentrate in the migration pattern of the agents. As we know, every agent is provided with a preferred list of migration sites (priority list of nodes for the agents) by their owner. Longer the priority list wider will be the agent’s scope to choose its next destination. But still, the probability of successful agent migration remains highly dependent on the position and connectivity of the next destination. Hence as shown in Figure 4.8(b), only a little improvement can be observed for longer priority list.

Keeping all parameters fixed if we increase the simulation time, the MANET diameter
4. Agent Reliability Estimate

Figure 4.9: Timely variation of reliability

increases and thus, overall reliability of MAS decreases. But after some time as network connectivity somewhat stabilizes, the rate of change of reliability with time reduces and the system enters a more or less stable state (shown in Figure 4.9).

4.7 Conclusion

The results obtained illustrate how environmental factors impact the network and hence reliability of MAS. This is important because it provides multiple possible solutions to meet an overall MAS reliability requirement. For example if nodes move with comparable speeds (lower relative mobility), MAS reliability is expected to increase. But to attain this same level of reliability spawning more agents is necessary in a volatile (higher relative mobility) or noisy MANET. The methodology presented here allows us to quantify the impact of the environmental parameters; which are not traditionally related to reliability.

These options provide the context for further analyses and the proposed methodology provides the foundations. It can be used to determine minimum MAS reliability characteristics for a typical MANET. Finally, once the network components and host platforms for the agents are developed, these methods may provide requirement verification where a reliability test of an entire network may prove too costly.