9.1 Introduction

In mobile wireless environments agents can be utilized for various purposes including service discovery [6], automatic network reconfiguration, clustering [60] etc. These agents can also be used to improve various other MANET services too. For example in [96] agents are used to place checkpoints close to the MNs for faster recovery. Agents can also be used to create intrusion detection systems for MANET [69]. But before large scale commercial deployment of such applications, availability and reliability and hence dependability of the agent system is needed to be analyzed. A dependable mobile agent based system implies that the agents will be available and reliable during the lifetime of the system.

In this context the application chosen here for such analysis is service discovery in MANET using mobile agents. Whenever a client application desires to access a service provided by a host or server, a service discovery procedure gets initiated. Although it is one of the fundamental applications that can run on MANET, the absence of any central intelligence in the network and limited energy of the MNs make the design and deployment particularly challenging. Many solutions are proposed but only a few have effectively addressed MANET uncertainties. Application scenarios for service discovery in wireless mobile environments are manifold

- In MANETs, some of the connected hosts might have, in addition to the ad hoc network interface, an external connection to the Internet. Such an ability can be regarded as a service for the participating ad hoc nodes. Using service discovery, members of the MANET are then able to use such a gateway service.

- Using their wireless hand-held device or notebook, participants in collaborative applications can discover application or game servers before participating in a session.

Any service discovery protocol is said to be well suited in MANET if

1. selection of a good instance of a service is possible and
Reliable Service Discovery in MANET

2. delivery of predictable services despite frequent topology changes

Reliability of the protocol will imply discovering the existence of good service in the network. Here comes the necessity of estimating reliability of such protocols. The higher the reliability of the protocol, greater is the probability of finding services in the network. Since agents can better tolerate MANET uncertainties, an agent based service discovery protocol is chosen for analysis. The service discovery protocol is described in [6]. The reliability is estimated following MC simulation technique that is developed in previous chapters. Then an agent based service discovery protocol is proposed that is designed to yield reliable performance despite network dynamics. The strategies used in designing the protocol can easily be extended in making other agent based applications reliable in MANET.

9.2 Service Discovery Process

A service in a network can be regarded as any hard- or software resource, which can be used by other clients. Service discovery is the process of locating services in a network. The following methods are used to discover and maintain service related data [66]:

- service providers flood the network with service advertisements;
- clients flood the network with messages in search for the desired service;
- nodes cache the service advertisements;
- nodes overhear in the network traffic and cache the interesting data (regarding services provided).

The first one corresponds to passive discovery (push model) whereas the next one describes active discovery (pull model). The other two methods mentioned above are the consequences of the first two. While the push mechanism is quite expensive in terms of network bandwidth (in the context of MANET), the pull mechanism suffers from poor performance (longer response times). Moreover there are other factors to be taken into account such as the size of the network (number of nodes), availability of a service (how frequently services appear and disappear in the network), and the rate of service requests. Traditionally static service brokers are used for sharing service information which is not suitable for MANET due to its inherent dynamic nature. So as in [6] mobile agents can be deployed for this purpose (looking for services offered) as the agents can migrate independently [127], behave intelligently [128] and negotiate with other agents according to a well defined asynchronous protocol [129].
The service discovery protocol presented in [6] is considered to be the basic protocol in this work. At first the reliability of MAS where agents are roaming around the underlying MANET, discovering various services provided by the nodes in MANET, is estimated. The algorithm in [6] uses two types of agents - a static Stationery Agent (SA) and mobile Travel Agent (TA). The SAs are deployed per node basis. On the contrary the TAs are deployed dynamically to collect and spread service information among the nodes in MANET. A TA prefers those nodes on its route which it has not yet visited but which are reachable via nodes it already knows. In order to enforce this TA Route algorithm is proposed in [6] that determines the next target migration site of a TA. The SAs are responsible for controlling the number of TAs roaming around the network. Thus depending on the incoming agent frequency (number of TAs visiting a node is said to be incoming agent frequency) of TA, an SA can either create or terminate a TA depending on network bandwidth. Larger the bandwidth more agents can be supported leading to better performance and probably improved reliability.

As can be seen from both approaches, if the agents are intelligent enough to learn and predict about node and link behavior, they can overcome the dynamics and uncertainties associated with MANET. Thus, by making MAS dependable, MANET is also made dependable.

### 9.3 Reliability Analysis of Service Discovery Protocol

In this work, we assume that MAS (S) at a time instant t has m(t) independent agents (Travel Agents in [6]) that may move in the underlying MANET. Here m(t) indicates the fact that the number of TAs varies with time as an SA can kill TAs [6]. The reliability of S is defined in terms of the probability of S being operational during a period of time [1]. Agent reliability is defined using equation 4.8.

The underlying network is modelled as an undirected graph $G= (V,E)$ as described in Section 4.2.1. Let the network consist of N nodes, thus $|V|=N$ that may or may not be connected via bidirectional links (e). Depending on a given probability a link e may either exist or not at a particular point of time. The node mobility is simulated using NHPP distribution following nhppMANET() in Algorithm 5.

In this scenario an agent can be viewed as a program visiting one node to another in the network (if the nodes are connected) based on the strategy listed as TA Route Algorithm in [6]. Here each TA carries with it a list of visited nodes called the Traffic Route Map (TRM). For each visited MN$_i$ in the TRM, a Route Sequence Number (RSN) is assigned and list of neighbors of MN$_i$ (LN) are also saved. The RSN counts the number
of previous visits of the TAs. With the help of the RSN the TA knows which nodes it has visited recently. The LN of a node contains all nodes that it can directly reach and which are yet unvisited by the TA \[6\]. If a TA visits a new node the id of this node will be removed from all existing LNs in the TRM. So LN can be used to identify nodes with unvisited neighbors and common neighbors of nodes. An example of TA migration is shown in Figure 9.1. As shown in Figure 9.1(a) a TA, first visits MN_1 and then MN_4. If the TA is then moving to MN_2 it contains a TRM as depicted in Figure 9.1(b). The RSNs show the order in which the TA has visited the nodes. The LNs contain all nodes that a node can directly reach except those which are already visited. For example MN_4 is a relay node that can communicate with MN_2, MN_3, MN_1, MN_5 and MN_6. Because MN_1, MN_2 and MN_4 are already visited they are deleted in the LN of all nodes in the TRM.

However in this protocol \[6\] node mobility in between agents’ journey is not considered. So necessary modifications are made to make the service discovery process more suited to the dynamics of MANET. A TA starts its journey from an owner (where it is created by SA) and moves from one node to another according to the TA route Algorithm \[6\]. But this movement is successful if the two nodes are connected and there is no transient error. So, a probability is associated with the movement to indicate transient characteristics of the environment. This is because 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 MN_A decides to move to MN_B (connected to MN_A) then the agent successfully moves to MN_B with probability \( p_t \). If at any time an agent finds all unvisited nodes to be unreachable, the agent waits and then retries. This step tolerates the transient faults (temporary link failure) as an agent retries after some delay and hence ultimately improves overall system
performance. This is not considered in [6] but to make the service discovery process more suitable to MANET dynamicity, transient fault tolerance becomes a necessity.

The algorithm followed by the TAs, TARouteMod() is listed in Algorithm 22. As

Algorithm 22 : TARouteMod(). An algorithm for choosing the next destination while being en route.

Input: List of nodes to visit (N here)
Output: Next destination to migrate to

begin
1 Find yet unvisited nodes, which are common neighbors of previously visited nodes.
   // Common neighbors have highest priority as their services can be used directly by more than one node.
2 if all common neighbors have been visited then
3   Visit the nodes not visited yet.
4   if there are more than one unvisited neighbors then
5     Visit any one of them.
6   if there are no unvisited nodes in direct range of the current host then
7     A node with unvisited neighbors is revisited.
8     if there are two such potential nodes then
9        The node with the lowest RSN is chosen.
10    else
11      The node with the lowest RSN becomes the node visited next.
end

such the agent’s decision in choosing the next destination depends on the currently reachable set of nodes. The assumption is that at short time intervals the network topology will not have huge changes so as to render the common neighboring nodes absolutely disconnected or be reduced to a node with very few neighbors. For example a TA may migrate to a node following step 6 of TARouteMode() (Algorithm 22) only to find that all unvisited neighbors of the node has gone out of range.

9.3.1 Reliability Analysis with MANET Following NHPP Model

The reliability of MAS (consisting of the TAs) with respect to the network status and its conditions (for example connectivity of the links, path loss probability etc.) is analyzed. Each agent is expected to visit all operating nodes in MANET in order to collect and spread service information. As mentioned earlier, transient failure is also considered. As such, for simulation purposes, we have assumed the failure probability (P) of the mobile nodes (P_node) to be a variable of Weibull distribution[29]. The MC simulation algorithm ServiceReliability() for estimating agent reliability is listed in Algorithm 23
Algorithm 23: ServiceReliability(). An algorithm for estimating reliability of the service discovery process.

**Input:** \( G = (V, E) \)  
**Output:** \( R_s \)

\[
\text{begin} \\
1 \quad \text{Steps 1-7 of BasicAgentReliabilityEstimation()} \text{ listed in Algorithm 4.} \\
2 \quad \text{Agent selects next destination following TARouteMod()} \text{ (Algorithm 22).} \\
3 \quad \text{Steps 11-16 of BasicAgentReliabilityEstimation()} \text{ listed in Algorithm 4.} \\
\text{end}
\]

9.3.2 Reliability Analysis of MANET with Mobility Models

Modeling MANET links with NHPP does not, however, effectively address relative mobility. But, in practice, relative speed of the nodes mostly decide about link existence. Hence mobility models may be used to simulate node mobility in different application scenario. Many mobility models are proposed that can address this issue based on a specific application scenario such as disaster relief or military movement etc. Here three mobility models are considered

- Random Waypoint Mobility Model [36]
- Smooth Random Mobility Model [100]
- Reference Point Group Mobility Model [130]

9.3.2.1 Random Waypoint Mobility Model (RWMM)

RWMM [36] is widely used to simulate node mobility for its simplicity. Here a mobile node randomly chooses a destination point (waypoint) in the area and moves with constant speed on a straight line to this point. After waiting for a certain pause time, it chooses a new destination and speed, moves with constant speed to this destination, and so on. Here we have chosen a linear velocity \( v_i(t) \) and a direction \( \phi_i(t) \) and calculated the next destination point as

\[
v_i(t) = \text{Random}\{0..v_{\text{max}}\} \quad (9.1)
\]

\[
\phi_i(t) = \text{Random}\{0..2\pi\} \quad (9.2)
\]
9. Reliable Service Discovery in MANET

\[ x_i(t + \Delta t) = x_i(t) + \Delta t \times v_i(t) \times \cos \phi_i(t) \]  
(9.3)

\[ y_i(t + \Delta t) = y_i(t) + \Delta t \times v_i(t) \times \sin \phi_i(t) \]  
(9.4)

But RWMM can result in a sharp turn or sudden stop when the differential rate of change of velocity is infinity, which is not feasible in a practical scenario [100].

9.3.2.2 Smooth Random Mobility Model (SRMM)

Modeling MANET with SRMM is already discussed in detail in Section 4.2.2. SRMM attempts to support temporal dependency in a way that the speed and/or direction of a MN at a time instant depends on the corresponding values at previous time instants. Though this is quite realistic, however, some application scenario like campus movements or disaster relief, users usually move in a group. Thus movement of one user affects the other. Such spatial dependency is not taken care of in SRMM.

9.3.2.3 Reference Point Group Mobility Model (RPGM)

In situations like battlefield or rescue work people work in groups and hence movement in a group is commonly observed in such situations. So we can also use RPGM [130]. In that case the speed and direction of mobile nodes (MNs) would follow that of their leader, called a reference point. The velocity of the leader can follow RWMM again. If \( v_{\text{leader}} \) and \( \phi_{\text{leader}} \) represent the speed and direction of movement of the reference point respectively then the speed \( (v_i) \) and direction \( (\phi_i) \) of MN\(_i\) can be calculated as follows [130]

\[ v_i(t) = v_{\text{leader}}(t) + \text{random}() \times SDR \times v_{\text{max}} \]  
(9.5)

\[ \phi_i(t) = \phi_{\text{leader}}(t) + \text{random}() \times ADR \times \phi_{\text{max}} \]  
(9.6)

Here SDR and ADR are speed and angle deviation ratio respectively having the following relation \( 0 < SDR, ADR < 1 \). Hence position of MN\(_i\) at \((t + \Delta t)\) time instant is estimated following equations 9.3 and 9.4.

Thus in the present analysis, the movement of the nodes is simulated in one of the above three ways. The received signal power is calculated according to two ray propagation [103] of radio signals as discussed in Section 4.2.2.
9. Reliable Service Discovery in MANET

9.3.2.4 Steps for Reliability Analysis

With mobility models and two ray propagation model in place, MANET can be formed. To assess agent reliability, equations 4.8 and 4.9 are used. However, the performance of the service discovery protocol is measured by a node’s success in discovering service providers in the network. All or a few nodes in MANET may act as service providers, thus total number of service providers, \( SP_{total} \leq N \). The instantaneous reliability of service discovery process spawned at MN\(_i\) \( (R_i(t)) \) can be given as

\[
R_i(t) = \frac{SP^{dis}_i}{SP_{total}} \tag{9.7}
\]

Here \( SP^{dis}_i \) denotes the number of service providers discovered by MN\(_i\). If \( R_i(t) \) is integrated over time, it gives the overall service discovery reliability \( R^i_{service} \) as follows

\[
R^i_{service} = \frac{1}{Q} \sum_{Q} \frac{1}{T} \int_0^T R_i(t) dt \\
\approx \frac{1}{Q} \sum_{Q} \frac{1}{T} \sum T R_i(t) \tag{9.8}
\]

Here time is divided in discrete time steps and snapshot of the system is taken in short periodic intervals. As this interval tends to 0 a near continuous view of the system can be observed. The same is repeated \( Q \) times according to MC simulation [102]. The MC simulation algorithm for estimating agent reliability, ExtendedServiceReliability() is listed in Algorithm 24. But these algorithms do not address various issues of MANET and therefore, may not always ensure service reliability.

9.4 A Reliable Service Discovery Protocol for MANET

In this section a service discovery protocol is designed in a way that achieves highest MAS reliability possible under a given MANET configuration. Reliability is estimated according to the Monte Carlo simulation process developed in [4] that considers QoS (link capacity) requirements of the agents.

Each mobile agent may have different QoS (link capacity is considered only) requirement [4]. So if there is a path between the current position and next destination of a mobile agent that satisfies its QoS requirement and both the nodes are operational, then only the agent will visit its next destination taking that path. Demanded link capacity can be simulated using Normal distribution as in [4]. So each agent having different QoS requirements will interpret the network in its own way.
Algorithm 24: ExtendedServiceReliability(). An algorithm for estimating reliability of the service discovery process.

Input: Initial position, initial speed and maximum acceleration of each node
Output: \( R_s, R_{i, service} \)

begin
1 Step 1-3 of AgentReliabilityEstimation() listed in Algorithm 6
2 Simulate mobility and find connected components using srmmMANET() in Algorithm 7.
   // RWMM or RPGM can also be used to calculate \((x_i(t), y_i(t))\)
   // in step 1-3 of srmmMANET() in Algorithm 7
3 for all agents \((M)\) in the system
4 if there is no software fault at the agent
5 Agent selects next destination following TARouteMod() (Algorithm 22).
6 Steps 9-14 of AgentReliabilityEstimation() listed in Algorithm 6.
7 if agent comes back to owner \(MN_i\)
8 Calculate \(R_i(t)\) using equation 9.7.
9 Spawn new agents if the desired service information is still unknown.
10 Steps 15-18 of AgentReliabilityEstimation() listed in Algorithm 6.
11 Calculate \(R_{i, service}^i\) using equation 9.8
end
In this scenario M agents from G groups start their journey from G nodes that act as their owners. Each owner equally divides among its agents the job of discovering services. Thus each agent gets a list of nodes whose information is needed by the owner. The agent may choose its next destination randomly from the connected nodes in its list. The nodes exchange their service information with the visiting agents - this can be done via message passing or through some stationery agent. The agents share their collected information with the nodes they visit. Hence a node may also come to know about a service provided by some other node and share its information with the visiting agents. Consequently, an agent, even without visiting a service provider node, may know about it and spread this information to others.

A timestamp is given to each service record in order to correctly predict the availability of services. While visiting a node if an agent finds newer service information about a service provider then the agent’s database is updated accordingly. Thus if an agent visits a service provider node then it updates its list accurately with the timestamp indicating how long the service will be available. But if an agent comes to know about a service provider from one of its visited nodes then it can only predict about the service. Thus indirect information must be given lesser weightage than that of direct observation, as prediction about a node is more accurate when an agent actually visits that node.

Moreover, in order to prevent the network being flooded with service discovery agents, every node has a predefined fixed agent tolerance limit. If this limit exceeds, the node starts killing its visitors, i.e., agents that come visiting. Thus an agent deployed by owner may not always come back, so a timer is used to check for such events. Whenever timeout occurs, binary exponential backoff algorithm [104] is used to create new agents by the owner node.

From equation 4.7 we know that MAS reliability depends on the reliability of the underlying MANET. The way working conditions at nodes and their connectivity \( R_{MANET} \) affects MAS reliability by affecting values of \( \lambda_{dir} \) and \( \lambda_{ind} \) is as follows.

If an agent, is asked to visit \( \lambda_{total} \) nodes, and it has ended up visiting \( \lambda_{dir} \) nodes directly and heard information about \( \lambda_{ind} \) nodes where \( \lambda_{total} \geq (\lambda_{dir} + \lambda_{ind}) \), then agent reliability is calculated as

\[
\lambda_i(t) = \frac{\lambda_{dir}}{\lambda_{total}} + \left( \frac{1}{\lambda_{ind}} \sum_{i=1}^{\lambda_{ind}} \text{decay}_i(t) \times \frac{\lambda_{ind}}{\lambda_{total}} \right) \tag{9.9}
\]

Here remaining time for which service at \( i^{th} \) node is valid is denoted by \( t_i \). We have
taken the decay function, \( \text{decay}(t) \) as shown below

\[
\text{decay}_i(t) = \frac{t_i}{t_i + 1}
\]  
(9.10)

Thus if a node is found to provide service for a longer time, then chances are that the service is more stable and hence the owner (and the nodes which are informed about this service by the agent) may get it if needed. Thus \( \lambda_i \) is much influenced by the conditions of underlying MANET like node mobility, interference, available link capacity etc. Now, the probability that the MAS is operational can be calculated as the mean of reliability of all its components (which are agents) in this system as follows

\[
R_s(t) = \frac{1}{m(t)} \sum_{i=1}^{m(t)} \lambda_i(t) r_i(t)
\]
(9.11)

This equation follows from equation 4.9. Here \( m(t) \) represents the number of agents present in the system till time \( t \), hence is not a constant (\( M \) in equation 4.9). In the above equation \( r_i(t) \) represents the probability that agent \( i \) is operational during time \( t \).

However the performance of the service discovery protocol is measured by a node’s success in discovering service providers in the network. All or a few nodes in MANET may act as service providers thus \( \text{SP}_{\text{total}} \leq N \). The instantaneous reliability \( R_i(t) \) can be calculated following equation 9.7.

If \( R_i(t) \) is integrated over time and the same is repeated \( Q \) times, it gives the overall service discovery reliability \( R^i_{\text{service}} \) following equation 9.8.

The procedure for the discovery and spreading of service information among the nodes in MANET using the notion of mobile agent groups is listed in Algorithm 25 (ServiceDiscoveryAgent()). The reliability estimation of the nodes and updation of agent feedbacks are shown in ServiceDiscovery() listed in Algorithm 26. In addition reliability of the agent system is also calculated using equation 9.11 while equation 9.9 along with 9.10 calculate individual agent reliability.

It can be observed that a service provider node, even without spawning agents, may collect fair amount of information about other providers from its visitors. Moreover according to equation 9.10, finding a stable service makes an agent more reliable than finding a scarce but unstable one.
Algorithm 25: ServiceDiscoveryAgent(). An algorithm executed by each agent for choosing
the next destination while en’route.

Input: List of nodes to visit (≤ N here)
Output: Next destination to migrate to

begin
1     if information about all nodes in the list are collected then
2         return // move back to the owner
3     Randomly choose from the list a connected but unknown node to migrate to
4         // the path must fulfill the agent’s capacity demand
5     if such a destination is found then
6         The agent moves with a certain probability.
7     if movement is successful then // the agent does not get killed
8         Find and update this node’s view of other service providers in the network
9         if that information has a valid timestamp.
10       Share agent’s collected service information with the node.
end

9.5 Experimental Results

This chapter follows our convention mentioned in previous chapters and the simulation
program is written in Java. The initial positions of agents are given. All agents of the
same group start from the same node, designated to be the owner. At first experimental
results of analyzing reliability of the service discovery process in [6] (see Section 9.3) is
described. This is followed by the results of Section 9.4. The results show how much
reliable the proposed service discovery protocol is for a given initial MANET configura-
tion.

9.5.1 Service Discovery with MANET being Modeled by NHPP

The default values of parameters are listed in Table 9.1. Unless otherwise stated, the
parameters always take these default values. First an example is cited for detailed
understanding of the algorithms TARouteMod() (Algorithm 22) and ServiceReliability() (Algorithm 23) with respect to the data generated by our simulation program. In our
example five nodes are taken to form a network. Every (Δt =) 3 seconds the positions
of the nodes and hence the network connectivity graph is updated according to NHPP.
Five mobile agents are deployed by the five different owners (nodes) and they start
their journey from their owners. Thus agents 0, 1, 2, 3 and 4 start their journey from
nodes MN₀, MN₁, MN₂, MN₃ and MN₄ respectively and roam around the network to
accomplish their task. Our job is to find the number of nodes that are successfully visited
Algorithm 26: ServiceDiscovery(). An algorithm executed by each mobile node for discovering services and reliability estimation of the system.

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

**Output:** $R_{service}^i$

```plaintext
begin
1 Repeat Q times
2 Initialize $SP_{collected}$ (consisting of $\lambda_{dir}$ and $\lambda_{ind}$) to 0
3 for simulation time $T$
4   Simulate mobility and find connected components using MANETCapacity() in Algorithm 8.
5   for all agents $m(t)$ in the system
6     ServiceDiscoveryAgent() in Algorithm 25 gets called.
7     A timer is started.
8     if a node receives more agents than it can handle then
9        It kills the agent it has received last.
10    if an agent comes back in time
11       Its $SP_{collected}$ is used to find $\lambda_i(t)$ according to equations 9.9 and 9.10.
12 else if a time out occurs then
13     A new agent is deployed according to binary exponential backoff algorithm.
14    if an agent retracts back
15       Update MAS reliability following equation 9.11
16      if information about desired service cannot be found
17         Spawn new agents with a yet undiscovered list of nodes.
18   Calculate $R_{service}^i$ using equation 9.8
end
```
Table 9.1: Default Values for the Configuration

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mobility Model</td>
<td>SRMM</td>
</tr>
<tr>
<td>N</td>
<td>25</td>
</tr>
<tr>
<td>M</td>
<td>20</td>
</tr>
<tr>
<td>Maximum Agent Frequency Tolerated</td>
<td>20</td>
</tr>
<tr>
<td>Link Failure Probability</td>
<td>0.1</td>
</tr>
<tr>
<td>T</td>
<td>750min</td>
</tr>
<tr>
<td>Q</td>
<td>100</td>
</tr>
</tbody>
</table>

by these agents which indicates how many services the agent discovers and how far it spreads discovered service information in the MANET and consequently the reliability of the agents will be calculated. Average reliability of all agents taken over a certain time period for a number of simulations represents the reliability of the MAS despite the uncertainties of MANET. Our migration policy is adopted from [6] and is mentioned in Algorithm 22 indicating the fact that all reachable destinations may not be equally likely. As shown in Figure 9.2(a), at time instant \( t=t_0 \) agents 0, 1 and 3 are stuck at their owners however agents 2 and 4 move to their respective neighbors. Thus after \( \Delta t \) time interval, agents 2 and 4 can be found at MN_4 and MN_2 respectively. In the next time instant agents 0, 1 and 2 discover services provided by MN_1 and MN_0 (Figure 9.2(b)). But agent 2 and agent 4 do not migrate as they have already visited MN_2 and MN_4 respectively. In the next time instant the connectivity graph changes (Figure 9.2(c)) significantly, enabling the agents discover more services. In the next time instant the nodes become sparse. Only agent 3 makes a successful migration. Thus as the simulation ends, only agent 3 is found to visit 4 nodes successfully thus covering \( \frac{4}{5} \) portion of its task. All other agents seem to have completed only \( \frac{3}{5} \) of their task. Thus the resulting reliability comes out to be \( \frac{\frac{3}{5} \times 4\frac{4}{5}}{5} = 0.64 \). So it can be concluded that in a MANET where the link failure probability is as high as 0.4 and the nodes can accept maximum 5 service discovery agents within a specified time (that is, incoming frequency), an agent, on an average is able to cover 64% of the network.

In reality network dynamicity affects agent migration and hence the reliability of MAS is found to depend on its size (number of agents) particularly for bigger MANETs. This is shown in Figure 9.3(a). The results are taken for two different scenarios. In one of them, each node can receive maximum (maximum tolerated frequency) 20 agents (white columns), additional agents would get killed resulting in a drop in agent reliability. For
the other one, the maximum tolerated frequency is kept at 15. As long as each node receives ($\leq 15$) agents within a short time, all agents will be able to complete their job if there is no software failure in them and the underlying MANET is stable as shown in Figure 9.3(a). But if $M$ is increased any further, then agent reliability drops as it exceeds the maximum tolerated agent frequency. Higher the number of agents tolerated in the network greater will be the overall agent reliability. This justifies the left side of the graph in Figure 9.3(a).

Every MANET has a bandwidth limitation that in turn restricts the maximum value of $M$ during a period. Thus, it can be observed that there is a maximum incoming agent frequency supported by a node (Figure 9.3(b)). Higher value of this indicates greater bandwidth provided by MANET. So as expected, with higher bandwidth our MAS become more reliable. But it can be observed that with $M=20$ when the incoming agent frequency reaches above 16, the MAS reaches an almost steady state with overall
reliability of (around) 0.98. This gives the optimum value of bandwidth to be provided by MANET for this scenario.

Now if $N$ is increased, the overall reliability does not change appreciably as long as $M \leq 30$ is comparable to the agent frequency (=20) supported by the nodes (Figure 9.4(a)). Thus our approach is found to be scalable for MANETs as big as 40 nodes. But for large $M > 30$, nodes may kill some agents resulting in a drop in reliability for $N > 32$. But this result indicates the scalability of the service discovery approach for crowded MANET as change in network size does not appreciably affect the reliability of
9. Reliable Service Discovery in MANET

Stability of the solution is also studied. As time increases, more agents will be able to tolerate transient link failures and thus the chance to discover and spread service information is higher resulting in improvement of reliability. But the graph in Figure 9.4(b) also indicates an optimum point (=750 min onwards) after which the system reaches a steady state for our configuration (Table 9.1). This figure also indicates that in absence of software fault at the agents MAS reliability may approach the ideal case (=1).

The link failure probability also affects reliability of the agents. As more links fail, the network becomes partitioned resulting in a sharp fall in MAS reliability (shown in Figure 9.5). It can be observed that when the link failure probability reaches above 0.5, the network may be broken into several components and so, some agents may never reach the nodes residing in another component of the network.

Links fail not only when background noise increases. Rather the most common reason for a link failure is the increase in relative mobility of the nodes. This node mobility can be simulated in a more realistic manner using mobility models as the results in the subsequent section shows.

9.5.2 Service Discovery in MANET - Modeled with Mobility Models

The effect of mobility models on agent reliability is shown here. The other factors that influence reliability of service discovery agents are already analyzed in the previous section. The default values of parameters are listed in Table 9.2. Unless otherwise stated,
the parameters always take these default values. Detailed analysis of simulation results is shown. The movement pattern of the nodes is found to play an important role in many MANET applications as it affects link existence probability. For smaller MANETs, nodes moving randomly according to RWMM exhibits better MAS reliability as compared to SRMM or RPGM (Figure 9.6). The reason can be density wave (where the average number of neighbors for a particular node periodically fluctuates along with time) [36]. But as MANET grows, the performance of MAS becomes almost independent of the effect of different mobility models and hence movement pattern of the nodes. This result shows that the reliability prediction made in one scenario (say campus network following SRMM) is very much applicable in several other scenarios (including disaster relief following RPGM).

The reliability of service discovery protocol is analyzed according to equations 9.7 and
9. Reliable Service Discovery in MANET

9.8. The performance is measured from the viewpoint of MN. First the performance is measured with number of service providers (SP\textsubscript{total}). As SP\textsubscript{total} increases service availability increases and hence reliability of the protocol after initial perturbation reaches a steady state (Figure 9.7(a)). But the effect of service availability is more apt in MANETs with more transient disturbance. As the environmental conditions stabilize the agents perform more reliably discovering almost all service providers even when they are scarce.

The effect of MANET size on the performance of service discovery is also studied as shown in Figure 9.7(b). But as a MANET becomes more crowded (N>30) redundant paths to reach service providers emerge, even when providers are scarce. Here the total number of service providers is taken to be 13. The drop in reliability for LFP=0.5 when there are 25 to 30 nodes in MANET is because at this stage the newly added nodes increases the MANET boundary rather than making it more crowded. Thus the overall network connectivity worsens. But with stable environment (low transient errors) the service discovery protocol manages to discover most service providers.

Performance of service discovery protocol can improve even further if nodes share their knowledge about service providers with the visited agents. This is validated in the following section with respect to the protocol detailed in Section 9.4.

Figure 9.7: (a) Performance of service discovery protocol with increasing service providers; (b) Performance of service discovery protocol with increasing N
9.5.3 Analyzing Performance of Proposed Service Discovery Protocol

We take the help of an example and show the data generated by the simulation program and then the detailed analysis of simulation results is presented. We have taken an instance where there are 6 nodes and 3 agents in the network. The connectivity graph, agents and their movements are shown in Figure 9.8. Due to smooth movement of the nodes (according to SRMM) no drastic change can be observed in the connectivity graph in subsequent time instants. Connectivity of the nodes is calculated according to the demanded link capacity of the individual agents that is simulated according to a Normal distribution with mean 3 and standard deviation (SD) 0.1. Thus the agents’ demand for different demanded capacity results in different views of the same MANET in terms of node connectivity. The links that offer fairly good capacity and hence fulfill capacity demands of all 3 agents are shown by blue bold lines in Figure 9.8 (for example link between MN$_5$ and MN$_6$ in Figure 9.8(a)). The thin black lines (Figure 9.8) represent links that fulfill capacity demands of some of the agents. The agents are shown by callouts where a dotted callout represents previous position of the agent. Nodes MN$_2$ and MN$_4$ are assumed to be service providers and MN$_2$ also deploys agents for discovering services. After the first time instant agent 1 migrates from MN$_1$ to discover a service provider (MN$_2$) while agent 2 discovers another service provider node (MN$_4$) as shown in Figure 9.8(a). In the subsequent time instants (shown in Figure 9.8(b) agent 1 spreads its collected information to MN$_3$. Thus MN$_3$ (MN$_5$) learns about the services provided by MN$_2$ (MN$_4$) from agent 1(2). In the subsequent time interval agents migrate even
Table 9.3: Default Values of Simulation Parameters

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mobility Model</td>
<td>SRMM</td>
</tr>
<tr>
<td>M (At the beginning of an experiment)</td>
<td>25</td>
</tr>
<tr>
<td>N</td>
<td>25</td>
</tr>
<tr>
<td>G</td>
<td>5</td>
</tr>
<tr>
<td>Mean of demanded capacity</td>
<td>3</td>
</tr>
<tr>
<td>SD of demanded capacity</td>
<td>0.1</td>
</tr>
<tr>
<td>Number of Service providers (SP&lt;sub&gt;total&lt;/sub&gt;)</td>
<td>6</td>
</tr>
<tr>
<td>Signal transmission power</td>
<td>25dBm</td>
</tr>
<tr>
<td>Minimum required signal power</td>
<td>18dBm</td>
</tr>
<tr>
<td>Length of PL(L)</td>
<td>N</td>
</tr>
<tr>
<td>$\Delta t$</td>
<td>3 seconds</td>
</tr>
<tr>
<td>T</td>
<td>6hrs</td>
</tr>
<tr>
<td>Q</td>
<td>100</td>
</tr>
</tbody>
</table>

Further spreading service information in MANET. Interestingly MN<sub>6</sub> comes to know about both service providers as both agents (1 and 2) visit this node. The later visiting agent (agent 1) may know about the earlier agent’s (agent 2) information as is reflected in MN<sub>6</sub>’s database. But after the third time instant, agent 3 manages to discover a service provider node (MN<sub>4</sub>). This process goes on. Thus some nodes without spawning agents may come to know about service providers if they are visited by knowledgeable agents. Since three groups of agents are considered each having an agent, each agent needs to visit all other nodes in the network. But as group size increases an agent needs to cover lesser number of nodes in the network. Thus agent reliability is also expected to increase.

A series of experiments is done to validate the proposed protocol. The default values of the parameters are listed in Table 9.3. Any change in these values will be explicitly mentioned. First agent reliability with increasing M for G=5 and G=10 (shown in Figure 9.9(a)) is evaluated. A node creates a group of agents to discover services in the network where each agent of one group has the same QoS requirement. Thus there can be maximum G different kinds of agents in terms of QoS requirements. It can be observed that reliability increases for larger group sizes and eventually reaches a steady state. Thus for G=5, the optimum group size is 5 agents per group (M=25 in Figure
9. Reliable Service Discovery in MANET

9.9(a)). However reliability of agents eventually reaches a steady state irrespective of the number of agent groups. Thus a node may not spawn large number of service discovery agents and consume bandwidth; rather it may learn about the service providers from other agents as well.

Agent collaboration plays a crucial role when a service is scarce. Larger the MANET harder the job of the agents and hence less reliable they are, at least with group sizes as small as $G=5$. But the situation improves with increasing number of service providers (gradual rise in Figure 9.9(b)) irrespective of MANET size. Performance readily improves ($SP_{total}=6$, $N=35$ in Figure 9.9(b)) if the providers are easily reachable by at least some of the agents. Scalability can be inferred since increasing $N$ does not change the overall nature of the result presented as graph in Figure 9.9(b).

The fact that MAS with different kinds of agents eventually reach a steady state is again confirmed in the next experiment that varies the demanded capacities of the agents. This is done by varying mean and standard deviation (SD) of Normal distribution to generate demanded capacity in ServiceDiscovery() algorithm (listed in Algorithm 26). Here mean denotes the average demanded capacity of the agent groups. SD indicates how much different the agent groups are in terms of QoS demands. For a given mean as SD increases, the heterogeneity among the groups increases even further. But this does not affect the reliability of the MAS in reaching a steady state (shown in Figure 9.10(a)). But when the QoS requirement of the groups is comparable (small SD), MAS performs better if the average QoS demands of the agents is less.

Performance of the protocol (equation 9.7 and 9.8) vs capacity demands is shown

![Figure 9.9: (a) Reliability variation with increasing M; (b) Reliability variation with increasing number of SPs](image)
Figure 9.10: (a) Reliability variation with demanded capacity; (b) Protocol performance with demanded capacity

in Figure 9.10(b). The performance is measured at MN₁ where MN₁ is not considered to be a service provider itself. The result indicates that for higher values of SD (10 onwards) the performance can be well predicted for a given mean. As lower mean with lower SD demands less capacity, this combination creates reliable agents.

Figure 9.11: Performance of service discovery protocol with increasing SPs

Discovering a service becomes expensive if it is scarce and hence performance of the service discovery then depends largely on relative position of the service provider with respect to the initiating node (that spawned service discovery agents) or the number of neighbors (of the initiating node) discovering the service and how well the initiating node is connected to its neighbors in terms of link capacity (Figure 9.11). For smaller values
of $SP_{\text{total}}$ the same argument applies. But as $SP_{\text{total}}$ becomes comparable to $N$, the situation improves and performance at a node (MN$_1$ here) reaches almost steady state. For larger MANETs scarce services can become highly expensive. In such scenario the information exchange by a node with its visiting agent can play a key role.

### 9.6 Conclusion

The basic idea of reliability estimation of agent based distributed applications on MANET is applied to service discovery process. Here the agents are deployed for collecting and spreading service information in the network. The reliability is found to depend heavily on MANET dynamics in terms of transient link failure and supported network bandwidth. However the effect of different movement patterns of the nodes is found to affect MAS reliability a little. Thus conclusion drawn for one scenario (represented by a mobility model) may well be applied to others (other mobility models) if the transient failure probability $<0.3$. The performance of the protocol measured through $R_i^{\text{service}}$ shows how agents help in discovering services overcoming the effect of poor network connectivity.

Moreover, a reliable service discovery protocol is designed using mobile agents that can tolerate much MANET uncertainties. A node acting as an owner creates a number of agents having similar QoS requirements and gives each of them different node list to visit in order to cover the entire network. The agents share their collected information with the nodes they visit and collect yet unknown information from them. A decay function is introduced for the information that an agent collects indirectly. Thus agent reliability increases if a stable service is discovered. The protocol is validated and results are shown in Section 9.5.3. The results indicate that the protocol achieves manageable reliability even when a service is scarce. Since the nodes share information with the visiting agents consistent knowledge base can be attained faster across the network. Agents are shown to perform reliably and attain steady state even when they (agents) are configured with different QoS requirement.