Chapter 10

Conclusions and Future Work

10.1 Introduction

The technology of mobile computing enables access to digital resources at any time, from any location [12]. Therefore distributed systems built on top of mobile wireless networks are characterized by motion and location independence. Though these features are extremely useful these affect both availability and reliability of the services of distributed systems due to constraints like limited resources, low bandwidth and node mobility. Because of its constraints the need for dependable services is even more intense. The existing systems offer only partial solutions, and often the approaches separate the issues of reliability, availability, security etc.

The thesis concludes with a review of the work presented in this thesis. The significance of the major results is summarised. Outstanding issues are discussed and directions for future work are suggested.

10.2 Summary of Contribution

This thesis concentrates on dependability analysis of mobile computing systems and aims at providing a framework for dependability evaluation of distributed systems running on wireless networks (with or without infrastructure) with mobile nodes. The work also investigates ways of achieving higher dependability.

Mobility, in wireless networks, induces two main factors in reliability estimation which are not present in wired network, namely handoff and intermittent connection to the network with short periods of disconnection (described in Chapter 3). Hence for estimating reliability in infrastructured network the following metric is considered: network coverage, two terminal reliability, all operating terminal reliability [28] and k-terminal reliability [29]. As shown in Chapter 3, if the distributed system can tolerate frequent handoff and longer disconnection intervals, the two terminal reliability improves and gradually reaches a steady state. Node failure is categorized into two types - recoverable (common) and irrecoverable (rare). It is seen that introduction of fault tolerance proves beneficial in the long run.
10. Conclusions and Future Work

Instead of message passing, mobile software agents can be used to coordinate distributed activities. This seems to be particularly useful when continuous connection is impractical and/or bandwidth is scarce as in wireless networks. The effectiveness of agent paradigm over message passing is shown in Chapter 3 in terms of system reliability in infrastructured network. Though MAS performance gradually reaches a steady state, for smaller number of MSSs (resulting in poor coverage) application run time seems to be critical for agent reliability. The results reported in Section 3.3.3 indicate that the MAS performs best when half of the nodes that an agent needs to visit are given higher priorities over the others. The agents are also shown to overcome frequent network disconnection.

Behaviour of an agent while en’route depends on the distributed application for which the agent is spawned. But agent performance in MANET is highly influenced by the dynamicity of the underlying environment. For instance, in MANET, each node is responsible for relaying packets to other nodes. So we analyze the effect of environmental factors on MAS performance in Chapter 4 that shows how MAS parameters may be configured in order to obtain the desired performance level in an unreliable environment. For example, in a relatively stable (lower relative mobility) MANET, MAS reliability is expected to increase. But in a highly volatile (higher relative mobility) or noisy MANET, spawning more agents is necessary to attain this same level of reliability. The methodology presented in Chapter 4 quantifies the impact of those environmental parameters that 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.

MAS can also be considered to be a system consisting of different agent groups where each group of agents has its own QoS requirements in terms of minimum link capacity or maximum tolerated overall delay. In Chapter 5, the effect of QoS requirements of the agents is analyzed with respect to MAS reliability. The reliability estimate is found to depend on QoS requirements of the agents, transient characteristics of the network and node mobility. It is observed that reliability approaches a steady state with different applications spawning agents having differing capacity demands. It is also found that agents with high delay tolerance limit can handle sudden noise bursts. If network condition is more or less stable, then capacity demanded by most of the agents is fulfilled. Since agents demanding for link capacity tends to follow efficient (more capacity lesser transient faults) routes, MAS reliability is found to be better as compared to the agents designed to tolerate longer delays. But as MANET diameter increases resulting in poor link capacity, delay tolerance of the application plays an important role in agent reliability.
MANET dynamicity makes it prone to security threats. When a mobile agent visits a host site, it runs as a process in that host platform and hence is dependent on that platform. So securing an agent while it is visiting a host is challenging. In Chapter 6 possible strategies for securing mobile agents in MANET are investigated. The concept of hashing is used to protect the agents against possible threats of modification of agent data and/or code by compromised nodes. The work reported in this chapter attempts to find distributed trust model for the network so that each trusted node may eventually get a consistent view of trust level of the network and hence prevent agents deployed by them from visiting compromised nodes any further. The success of this reputation scheme is measured in terms of Ratio of Agents Passed(t), Ratio of Successful Agents(t), Node Success Ratio(t) and Ratio of False Negatives. The trust based reputation models proposed in this chapter enable an agent to share information about the nodes it suspects, helping MNs update their trust levels. This enables faster convergence of trust view and nodes visited by an agent can also come to know about MANET hostilities without deploying agents themselves. Modification of agent code and/or data in transit is also detected eventually. Thus any change in node behaviour can be eventually detected. Though for a highly noisy environment, longer time is needed for the trust views of the nodes in MANET to reach a steady state. It is found that if bandwidth is sufficient to support agent migration, irrespective of network size, the extended trust model described in Section 6.4.3 of Chapter 6 can detect all malicious nodes resulting in zero false negatives. This indicates the correctness of the model.

Not only security issues, other issues like selfishness that results from the constraints of MANET (due to motion and location independence) is analyzed in Chapter 7. A trust based reputation scheme following Dempster-Shafer belief theory is developed that prevents the mobile agents and nodes from communicating with the misbehaving nodes that covers both malicious and selfish ones. The scheme puts an upper bound on the maximum number of agents spawned irrespective of network conditions. The effect of this scheme on agent reliability is then analyzed critically. It is found that with growing number of misbehaving nodes, agent reliability gradually falls. But the proposed reputation system can prevent sharp fall in reliability even when there are significant number of misbehaving nodes (>40%). This shows the proper utilization of partial response (sent by the agents while enroute) despite minimum bandwidth overhead. It is found that an early change in node behaviour causes greater reduction in reliability than if the same comes later. Comparison between the reputation schemes reported in chapters 6 and 7 indicate that the scheme described later (Chapter 7) outperforms the other for a hostile MANET as it uses partial response which makes faster detection of
Designers of distributed applications for MANET are often faced with a design choice of whether to use agents or to go for simple message passing. The typical advantage of using agents is that all distributed hosts do not need to be synchronized or even connected at one time. But in a hostile MANET where some nodes may not behave rationally, message passing and agent migration are equally affected. Performance degrades since some agents are occasionally dropped. Thus in Section 7.5 the effect of growing number of misbehaving nodes is studied on MANET reliability, reputation scheme and on agent reliability. The metric used to measure MANET reliability is k-terminal reliability. It is found that in a hostile MANET with falling KTR, applications deploying agents can show some progress always because of the reputation scheme, thereby increasing the effective reliability of the distributed application. The reliability improvement (of distributed applications spawning agents) is more significant when there are more nodes that spawn agents with differing migration trails. Thus the effect of securing agents results in improving its reliability as well.

Reliability and security are two important attributes of dependability. If a system is reliable and secure then it is expected to be dependable as well. In Chapter 8, dependability of an agent based system on MANET is quantified. It is viewed as a combination of availability and reliability while security affects both. Such a composite measure can more effectively assist the application designers in specifying the requirements of applications deploying agents, for example, what should be the average length of priority list (list of nodes an agent is asked to visit), how to have both suspicious and trusted nodes in an agents priority list without losing agent dependability, when to create a new agent etc. System dependability is an important metric for evaluating system performance particularly in unreliable dynamic environments like MANET. This governs agent design, deployment and migration strategy of the agent based system. Since the environment is dynamic, a dynamically customizable strategy could be beneficial. Here comes the need for reinforcement learning. On-policy Monte Carlo method [5] of reinforcement learning is applied to help the agents better avoid suspicious nodes in MANET. It is found that even with no learning the agents perform better when 66% of the nodes deploy more agents than others. However learning shows better dependability over its counterpart and as MAS gets bigger, this effect is more profound. Moreover from the results reported in Section 8.4, it can be concluded that, in a hostile MANET, agents should be designed with smaller trails and more conservative behavior (always avoid a suspicious node).

This approach indicates that if the agents are intelligent enough to learn and predict about node behavior, they can overcome the dynamics and uncertainties associated with
MANET. Thus, by making MAS dependable, MANET also becomes dependable.

The basic idea of agent reliability estimation developed in chapters 4 and 5 is applied to service discovery process described in [6]. It is found that with transient failure probability \(<0.3\) results representing a specific mobility model may well be applied to other mobility models as the movement pattern of the nodes hardly affect MAS reliability. Reliability is found to improve as the network supports more number of agents. A reliable service discovery protocol using mobile agents is designed that can tolerate MANET uncertainties. 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. Since the nodes share information with the visiting agents consistent knowledge base can be attained faster across the network. This collaboration is found to play crucial role specially when a service is scarce. System performance readily improves if the providers are easily reachable by at least some of the agents. Agents are shown to perform reliably and attain steady state even when they (agents) are configured with different QoS requirement.

Thus this thesis provides methods for quantifying and analyzing the dependability of agent based distributed system on MANET. Effect of environmental parameters on reliability and hence dependability analysis of the system is shown which are not encountered in traditional reliability analysis. It is found that though the distributed applications that spawn agents do not depend on each other, MAS performance depends on the variation of agent characteristics in terms of QoS demands and/or node lists to be visited. More variation results in better overall performance particularly when MANET is hostile. This leads to an important observation that different agent based applications do not hamper system performance rather improves reliability even further. The methods documented here are useful to the system engineer because they enable the analysis of system dependability when certain system and node attributes such as node reliability, node count, QoS demands of agents etc. are known. With the methods reported in the work the designer is now capable of setting and manually optimizing these input parameters such that reliability objectives are met. 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.

### 10.3 Outstanding Issues

The simulation results reported in previous chapters indicate that secure agent based system effectively improves the dependability of distributed applications despite network
dynamicity. Dependability further improves when the agents dynamically learn to avoid malicious nodes and/or nodes connected via erroneous links. Nevertheless, the framework has some limitations. This section discusses these limitations and suggests that further work is necessary to resolve these issues.

- The present work deals with independent agents. But there are applications where mobile agents may not be independent of each other. For example, agents used in intrusion detection systems. This is not taken care of by the reliability estimation model. But the model can be readily extended for this purpose.

- Conclusions drawn by our reliability as well as dependability estimation techniques are based on the assumption that nodes move according to Smooth Random Mobility Model [100] and two-ray propagation model [104]. But this may not always reflect many real application scenarios. To make the models reasonably accurate but simpler these models are adopted. Thus methods depicted in chapters 5 and 8 could be modified to include group mobility model [130].

- We defined dependability only by its technical aspects thus viewing it as a combination of availability, reliability and security. But other aspects like safety and maintainability have to be considered. However, safety plays a crucial role in safety critical applications and maintainability will be important while designing an agent's code.

- All experiments in this thesis are simulated using Java. Although there are a few mobile agent platforms available for example, IBM Aglet [131], Concordia [132], Grasshopper [132] etc., most of these are heavyweight and cannot be installed in resource-constrained devices like PDAs. Moreover, these platforms do not allow strong migration of the agents. However, we could not test our technique on real mobile devices but simulated it as simulation provides researchers with a number of significant benefits, including repeatable scenarios, isolation of parameters, and exploration of a variety of metrics.

10.4 Future Work

A variety of research problems arise from this thesis work. Some involve relaxing one or more of the modeling assumptions that have been made, and still others involve application of the methods developed to other distributed or networked System of Systems (SoS) for the evaluation of reliability or other system attributes that take on a proba-
10. Conclusions and Future Work

The technique developed in Chapter 7 for calculating KTR can be extended and applied to estimate reliability measures in Distributed Real-time Embedded Wireless Systems (DREW). Real-time and embedded systems are increasingly being connected via wireless and wireline networks to create large-scale distributed real-time and embedded (DRE) systems, such as tele-immersion environments, fly-by-wire aircraft, industrial process automation, and total ship computing environments [133]. As wireless distributed systems, DREW systems are inherently noisy and harsh as environmental noise affects system performance quite often even when the embedded nodes are static. One such example is that of a wireless sensor network. Such networks suffer from common cause failure which refers to some environmental condition affecting a cluster of components and individual component failures that can be assumed to be poissonian. We can say that a DREW system is reliable when some components failed due to common cause; in addition some paths needed by our real time application are not working due to individual component failure; still there is a path from a source to the destination that may well carry the real time traffic.

This is expressed mathematically as

\[
R(\text{System}) = \sum_{C_{cc} \in \Psi} P_{cc} \cdot \sum_{i=1}^{P'} (2TR_{\alpha_i} \times P(\alpha_i)) \tag{10.1}
\]

The terms used are summarized in Table 10.1.

<table>
<thead>
<tr>
<th>Terms</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(R(\text{system}))</td>
<td>System Reliability</td>
</tr>
<tr>
<td>(P_{cc})</td>
<td>Probability of Common Cause Outcome</td>
</tr>
<tr>
<td>(C_{cc})</td>
<td>A Common Cause Outcome</td>
</tr>
<tr>
<td>(P')</td>
<td>Number of paths not affected by a common cause</td>
</tr>
<tr>
<td>(2TR_{\alpha_i})</td>
<td>Two terminal reliability for (i^{th}) path</td>
</tr>
<tr>
<td>(\alpha_i)</td>
<td>A network configuration of the embedded nodes corresponding to (i^{th}) path</td>
</tr>
<tr>
<td>(\Psi)</td>
<td>Set of all common cause outcomes that can occur during mission time</td>
</tr>
</tbody>
</table>

Equation 10.1 sums over all possible probability of common cause (cc) failure. It says given a common cause failure (with probability \(P_{cc}\)) if there exists \(P'\) working paths not affected by cc then for each possible path the probability of successful transmission can be
predicted using two terminal reliability. Now reliability of DREW systems using equation 10.1 can be calculated. However it is found that analytical methods are appropriate for abstract models only. If (dynamic) network topologies have to be taken into account, for instance, when the embedded nodes move, analytical methods fail and the assessment of reliability measures has to be done numerically using some simulative methods. Monte Carlo simulation based algorithms can be designed towards evaluating reliability of DREW systems. By applying our methods developed in algorithms Algorithm 7 in Chapter 4 and Algorithm 1 in Chapter 3 reliability metric like 2TR, AoTR and Network Coverage for DREW systems can also be estimated. Thus our approach for reliability calculation can be applied to estimate reliability of DREW Systems.

Specific to the problem of dependability evaluation of mobile computing systems, some proposal for future work are given below:

- Machine learning techniques can be applied to optimize reliability of a distributed application in MANET with respect to network parameters like mobility, path loss, battery power of the nodes etc. However optimization techniques can be applied. Optimization techniques can even be applied to design the agents optimized for higher reliability for a given network scenario. This would provide a quantitative mechanism to design and prevent the hunt and peck approach to optimization that occurs in its absence.

- Energy efficiency of the devices can be taken care of while designing security related protocols. For instance while calculating reputation of a node remaining energy can be an additional parameter along with direct and indirect feedbacks of the agents as discussed in Chapter 6. Thus the final trust calculation of a node can be viewed as a weighted average of the trust computed from direct and indirect feedbacks and the remaining energy as is predicted of the node. The nodes may exchange remaining energy information through some MAC protocol or piggyback this information with route request packets from which using some decay function the remaining energy of a node can be predicted. Consequently it can be predicted when the node may start behaving selfishly.

Remaining energy not only improves agent and thus node security but also helps in better estimating availability of a node in MANET.

- Nowadays in MANET there is a growing demand for real time systems that need bandwidth guarantees. But as most of these systems are fault tolerant, these (the systems) provide some gracefully degraded states in presence of failures possibly
with less speed. In this context performability [21] is a metric that better measures performance of such systems. It may be viewed as a composite measure of dependability and performance, for systems that exhibit smooth degradation in performance in response to failures, ranging from fully operational, to slowly or partially operational, to completely failed. Performability recognizes the joint contributions of unavailability, unreliability and low performance, in determining the net level of service that is deliverable to clients of a distributed system. This can be applied to agent based mobile computing systems particularly for applications that have some time constraints.