ABSTRACT

THANGAVELU, KRITHIGA. Maximizing Service Coverage of Adaptive Services in Wireless Mobile Ad-Hoc Networks using Non-Clustering Approach. (Under the direction of Douglas S. Reeves).

Wireless Mobile Ad-hoc Networks are characterized by dynamic network topology and lack of network infrastructure. The network fragments into smaller networks and merges over a period of time due to mobility. This makes provisioning solutions to common network problems, like routing and QoS provisioning, a challenging task.

Services in ad-hoc networks face two-fold problems. Making nodes aware of the availability and the location of services in a dynamically changing network is difficult, especially when such services are not tightly coupled with a fixed infrastructure. Servers may come and leave the network. Nodes may shutdown services to conserve energy. The problem is further exacerbated by the limitations posed by the wireless network on the bandwidth and by the limited computational capability of the wireless devices. This thesis addresses the problem of providing continuous and guaranteed access to such centralized services in a mobile wireless ad-hoc network.

A distributed algorithm based on the exchange of service provider information is proposed to solve the problem. The previous work addressing the same problem assumes that the nodes move in long-term groups. Our solution does not make this assumption and targets arbitrary motion. So, no attempt is made to correlate the movement of the nodes, in order to solve the problem. In this thesis, we illustrate that our approach achieves higher service availability than the previous methods at the cost of a higher number of service instances.

The proposed algorithm converges after a time period equivalent to the average propagation delay of the service instance information from a service provider to its reachable nodes. The computational and communication complexity of the algorithm is theoretically proved to be O($s\log n$) and O(n_g^2) where s is the number of service instances, n is the number of nodes in the ad-hoc network and n_g is the average number of nodes in a connected component of the graph formed by the nodes in the ad-hoc network.

The service cost incurred in providing the necessary service coverage is proved through simulation to be in the order of the number of connected components in the graph formed by the nodes in the ad-hoc network. Simulation results are used to prove that the algorithm provides for maximum service coverage independent of the mobility pattern of the nodes in the ad-hoc network.

Maximizing Service Coverage of Adaptive Services in Wireless Mobile Ad-Hoc Networks using Non-Clustering Approach

by

Krithiga Thangavelu

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Master of Science

in

Computer Science

Raleigh
January 6, 2003

Approved By

Dr.Gregory T. Byrd
Dr. Munindar P. Singh

Dr. Douglas S. Reeves
Chair of Advisory Committee

To

My Wonderful Parents

BIOGRAPHY

Krithiga Thangavelu received her bachelor's degree in Computer Science and Engineering from Bharathiar University, Coimbatore, India in May 1999. She has been pursuing a master's degree in Computer Science at North Carolina State University, Raleigh, NC since the fall of 2000.

ACKNOWLEDGMENT

I would like to thank my advisor Dr. Douglas Reeves for his invaluable guidance, encouragement and for providing me the opportunity to work with him on my thesis. I would like to thank Dr. Gregory Byrd and Dr. Munindar Singh for agreeing to be on my thesis committee.

I'm immensely grateful to Dan Stevenson and TJ Smith of MCNC and Dr. Gregory Byrd of NCSU for giving me an opportunity to work with them on their research projects.

I'm indebted to my family for their understanding and support in all my endeavors. I sincerely appreciate the moral support offered by all my friends and colleagues whose constant reminders kept me moving in the direction of completing my masters program.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Mobile Wireless Ad-hoc Networks

Wireless technology is a fast emerging technology undergoing evolution day by day due to popular demand. One sphere of wireless technology that is highly addressed in today’s research is the mobile ad-hoc network. The advent of different kinds of mobile and portable devices has given rise to the desire to be able to communicate anytime, anywhere, without the aid of a centralized fixed infrastructure. This kind of networking is loosely termed as Ad-hoc networking. In short, ad-hoc networks are networks formed on the fly by a group of wireless devices in an ad-hoc fashion: infrastructure-less with no configured but varying network topology.

Ad-hoc networks pose a unique set of problems requiring that already solved network problems be readdressed. Most of the available network solutions are based on the existence of "fixed" reliable networks with rare occurrences of link failures and the existence of mobile components as appendages to the fixed infrastructure. Since ad-hoc networks are formed purely by mobile wireless devices, link formation and breakage are ubiquitous phenomena, calling for a new approach to solve common network problems like routing, QoS provisioning and Security. Yet, the scenarios where one does not want or cannot deploy and manage an infrastructure justify the need for ad-hoc networks.

The notion of ad-hoc networks can be associated with a wide spectrum of communication needs ranging from communication in combat fields and disaster recovery sites, to requirements in the commercial sector like sharing of information in conferences, communication between different appliances in a home network or among a group of robots doing security surveillance or environmental monitoring.

Ad-hoc networks are not only envisioned for communication between reasonably sized devices but are also speculated for use in intercommunications between miniature computational elements used for diagnosis and study of working of systems ranging from different forms of life to manufacturing systems. Ad-hoc networking is the only possible solution for communication between such devices and in cases where wired connectivity is infeasible for communication between a huge number of devices confined to a small area. Ad-hoc networks are envisioned to be smaller temporary networks than traditional networks.

1.2 Constraints posed by Mobile Ad-hoc Networks

A wireless ad-hoc network is a set of mobile or semi-mobile devices communicating with each other directly or indirectly with the help of intermediate nodes in the network. All the nodes in the network communicate through wireless links. Since not all the nodes in the network are within the transmission range of each other and no node is constrained to form the network crux, routing or forwarding capability is desired in all the nodes in the network.

Ad-hoc networks are expected to be highly tolerant to node and path failures. This feature is a consequence of the inherent need for nodes to be mobile and not be dependent on other nodes. Hence network functions have to run in a distributed fashion since the nodes might suddenly appear and disappear from the network.

Wireless networks have a major constraint on the resources like limited bandwidth with limited transmission range and very little computational capability of the nodes. A solution being proposed for an ad-hoc network problem has to be efficient enough to avoid wasting the limited resources.

When the nodes are mobile, links are formed and broken arbitrarily. As a result, these networks frequently break into a number of disjointed networks and merge over a period of time. Yet the connectivity in the network should be maintained for uninterrupted operation of applications and services.

Since the nodes in the network communicate through wireless links, the link capacity fluctuates with a high bit-error rate. This problem is further exacerbated by the mobility of nodes. Communication between the nodes should involve as few hops as possible to bind the transmission error rate.

Most devices are battery driven, so the power demand on the nodes should be minimized. This need directly translates to minimum usage of the resources of a node by other nodes in the network.

Since cooperating nodes form the network, there is no concept of centralized administration. Ad-hoc networks are required to operate in an environment where all or some nodes are mobile and yet provide for the same user level requirements on connectivity and traffic delivery as in wired networks.

1.3 Services in Ad-hoc networks

Applications in ad-hoc networks can follow one of the following paradigms: (1) network transparency and (2) mobility aware adaptation. Making applications oblivious to the changes in the environment comes with a compromise on the efficiency of the system and the application's performance . Providing application level transparency complicates the system design as the underlying infrastructure has to make adaptation decisions for diversified applications. Mobility-aware adaptation allows application to react to resource changes in a prompt manner. One way to make applications adapt to change in mobility is through collaboration of system and individual applications. System can monitor the resources and provide information to the application to make resource specific decisions. Mobility adaptation logic can exist on the clients or on both clients and servers or on a proxy in client-server models.

Services in ad-hoc networks can be classified into two types:

(1) Peer-to-peer services, and

(2) Centralized services.

The basic need for all the envisioned applications of ad-hoc networks (refer Sec.1.1) is information sharing. Ad-hoc networks are inherently peer-to-peer since no node in the ad-hoc network is assumed to be a part of the network during the entire time period of its existence. Some popular P2P systems currently available for fixed or stable networks are Gnutella[1], Napster[2] and JXTA[3]. Gnutella and Napster are specifically designed for sharing of files across the internet. In Gnutella, a file server joins the Gnutella network by notifying the neighboring Gnutella file server. When a query arrives at a file server, it attempts to answer the query or forwards the request to its neighboring Gnutella file servers. In Napster, all file servers register with a central server and provide information about the documents stored in the file servers. Clients query the central server to find file sources. The usage of these two peer technologies without adaptation in ad-hoc networks is not feasible as location of servers or nodes hosting the files, changes dynamically. Resource discovery and advertisement are essential design components to be defined for the above file sharing P2P systems to work in ad-hoc networks. Ad-hoc networks are inherently peer-to-peer since no node in the ad-hoc network is assumed to be a part of the network during the entire time period of its existence.

Yet all the services in an ad-hoc network cannot have a peer-to-peer architecture. Important network applications and services, like web servers, information databases and network management functions inherently follow the client-server paradigm. Since the topology of an ad-hoc network is dynamic, we cannot rely on any one node to provide the centralized service.

1.4 Service Provisioning in Mobile Ad-hoc Networks

As we've seen in the previous section, service provisioning in a network has to address the following issues:

- From a service provider's point of view, advertisement of available services and their location and
- From a client's point of view, dynamic discovery of the service and optimal server selection.

One popular method to advertise the location and availability of services is through a central registry with which all the service providers register their services. The drawback of this method in ad-hoc networks is the lack of a fixed infrastructure with which the location of the registry can be preconfigured. Mobility makes it impossible to provide continuous access to the registry for all the nodes. Another method to advertise service location is through broadcasting or multicasting. Broadcasting and multicasting of information results in a larger consumption of bandwidth compared to that in wired networks. Moreover, due to mobility of the services themselves, such broadcasting sessions have to be periodic to be of any use.

Nodes in an ad-hoc network come and leave the network anytime. Servers host services depending on the availability of resources. They might shutdown services to conserve energy and for efficient use of the computational capability. As a result, nodes in the ad-hoc network should be able to dynamically discover new services. Due to mobility and limited bandwidth in ad-hoc networks, nodes should access services closer to their location in order to have prolonged access to the services. Accessing services closer to the node results in a fewer number of hops and hence fewer transmissions, which means less possibility of network congestion.

1.5 Example

Large-scale networks of wireless sensors are becoming a reality with advances in the hardware technology design of miniature computational components[4, 5]. Smart Dust[4] are millimeter-scale nodes with sensor functionality and communication capability. Its current prototype devices are much larger than the smart dust elements (size of a sand particle) and communication is accomplished using RF transmissions over short distances. The communication networks envisioned for these devices are mobile wireless networks.

Imagine a military-based scenario where a large number of miniature mobile elements are dispersed in the war zone. These elements are assigned the task of collecting information for surveillance, about the presence of chemical or biological agents, in the battlefield. Mini-database servers are hosted on some of these elements to store the collected information. A small flying vehicle picks up the collected information. The number of database servers should be minimized to reduce the number of stopovers for the vehicle. An element hosting a database server has three tasks competing for the CPU time. They are

(1) the task of running the database,

(2) the task of receiving information collected by neighboring elements and

(3) exploration by the element. The element acts as a fullfledged database server depending on the number of clients. The task of hosting a database incurs a fixed cost whereas the task of receiving information is a variable cost factor depending on the number of clients. The lesser the number of clients, the more is the amount of CPU time available for exploration. We also need to keep the

number of databases to a small value (<<the total number of elements) inorder to reduce the fixed cost incurred by the network as a whole, while providing an accessible database server to all the elements. So, an element performs only exploration, as long as a database server is accessible and hosts a database server when isolated from database-hosting elements. When a previously isolated element is highly connected and with very few clients, it transfers the stored information to a running database server and becomes a full-fledged explorer. When a database server reaches its maximum storage capacity, it shutdowns its service and becomes an inactive participant (neither an explorer nor a storage provider). Since the mission undertaken is in a hostile environment, the flying vehicle does not pick information from arbitrary servers. The servers have to identify with the flying vehicle using a secret code. This code is changed during each visit. As a result, an element intending to host a server should obtain the code from a current database-hosting element. The same code is also used to transfer the collected information from one's server to another server.

To acheive this scenario, we need an infrastructure that supports dynamic service discovery, isolation prediction, adaptation to a mobile environment and a service provisioning algorithm that optimizes the number of database servers in the ad-hoc network formed by the elements while ensuring access to a database server for all elements. We will propose an application layer middleware to provide for the required infrastructure in this thesis.

This scenario is similar to the scenario addressed by the research problem in Wang and Li[6, 7] especially when nodes move in groups. Their research work addressed the problem of providing maximum service coverage at the cost of minimum number of service instances in ad-hoc networks. These services can be replicated by downloading service hosting information. A distributed QoS provisioning algorithm is proposed to do the same. In our example scenario, database servers, elements and secrete code are similar to the service instances, nodes and service hosting information of the problem[6, 7]. Since the information required to host the database is the secret code which is downloadable, we will refer to the database server as an adaptive service[6, 7] since the service can migrate from one node to another, be replicated when necessary and terminated dynamically. Service migration or termination is possible because we can transfer the information from one database server to another.

Since the movement of the elements in the example scenario is motivated by the mission rather than a group expedition, mobility will not be strictly in groups. Since the goal of these elements is to collect as much information as possible, their rate of movement varies depending on the need of the moment.

1.6 Motivation

Researchers have tried to predict node mobility and correlation in the movement of nodes in order to simplify the solutions to common network problems like routing and QoS provisioning in an otherwise ad-hoc network[8, 9, 10, 11, 12, 13, 14]. When nodes do not follow any pattern for a significant period of time, mobility pattern prediction schemes do not help to improve the performance of the strategies based on them. This is seen in the case of the solution[6, 7] proposed for the research problem addressed in this thesis when the nodal movement is not strictly a group oriented movement. Though movement prediction promises to improve efficiency, reliability and adaptability of wireless networks, it reduces the adaptability of the solutions to frequent changes in network conditions especially when the network is highly dynamic. This motivates us to use a non-clustering approach for solving this research problem.

Most performance studies and others reported in literature, have used simplified movement models that do not accurately characterize the user mobility and hence lead to unrealistic conclusions[14]. Trace based mobility models best characterize realistic network scenarios and are more accurate in protocol validation [15]. Since ad-hoc networks are yet to be widely deployed, an exact definition of realistic models of various scenarios is not yet available. Solutions that address the problem in a more generalized manner and do not rely heavily on mobility patterns in an ad-hoc network can be refined when trace based mobility models become available in the future in order to provide for pragmatic solutions.

Another major assumption made by most application level ad-hoc network researchers[6] is the accurate knowledge of all reachable nodes from a node in the network at any instant of time. When the network is significantly mobile, given the constraint of limited bandwidth and transmission range of the nodes, such an assumption is not a reasonable one.

When a clustering approach is used to solve the problems in ad-hoc networks, the quality of service offered to the nodes in the network is not uniform. Nodes whose movements have strong correlation with cluster movements receive the best quality of service compared to occasionally digressing nodes. This further warrants the use of a non-clustering approach that gives uniform quality of service to all nodes in the ad-hoc network, and is highly responsive to changes in the network topology.

Hence, we will attempt to provide a non-clustering based solution to the problem of providing continuous access to adaptive services for nodes in the wireless mobile ad-hoc networks.

1.7 Contribution

We make the following contributions in this study:

1. We introduce a non-clustering strategy for QoS provisioning of adaptive services in ad-hoc networks.

2. We improve the basic non-clustering strategy by using optimization heuristics that minimize the unnecessary change in the set of service instances and also minimize the frequency of change in the service provider accessed by a node.

3. We evaluate and compare the performance of the proposed algorithm against a Velocity based clustering algorithm proposed by Wang and Li[6, 7] and show that the newly proposed algorithm provides better service coverage when group mobility does not hold good in the nodal mobility pattern. This solution is different from the solution proposed by Wang and Li in that it does not attempt to detect patterns in the mobility of nodes but rather uses a modified version of flooding.

1.8 Thesis Organization

The thesis is organized as follows. In Chapter 2, we survey the research done so far in the field of ad-hoc networks and discuss the related work in QoS provisioning in ad-hoc networks. We then describe the system model on which this thesis is based and state the problem this work addresses. We then outline our approach to solve the problem. In Chapter 3, we discuss in detail the non-clustering adaptive service QoS provisioning algorithm and the rationale behind the various strategies incorporated in the algorithm. We then theoretically derive the complexity of the algorithm. In Chapter 4, we define the different performance evaluation metrics and outline the simulation environment in which the algorithm is evaluated. We then analyze the simulation results. Finally, we summarize our work in Chapter 5 and provide directions for future work.

Chapter 2

Background and The Problem

In this chapter, we explore the different approaches to solving the problems in mobile ad-hoc networks and explore how the application layer problems are approached by prior work in the field of ad-hoc networks. Then, we discuss the related research work and compare it with our work. Then, we briefly outline the research problem and the proposed strategy.

2.1 Background

Research in ad-hoc networks at the network and physical layer is comparatively more developed than other aspects or areas of the ad-hoc networks. Some of the optimization strategies suggested for routing can be adapted for solving other problems in ad-hoc networks.

2.1.1 Techniques to counteract the effects of Node Mobility in Ad-hoc Networks

The main problems posed by node mobility in ad-hoc networks are:

- Change in connectivity of the network
- Fluctuation in link capacity
- Network fragmentation and
- Difficulty in locating the position of a node in the ad-hoc network by another node.

Mobility Prediction

Mobility Prediction techniques is a class of techniques that addresses the problem of finding stable routes in a network with dynamic topology. Mobility Prediction techniques define path stability metrics based on the connectivity and longevity of the links between nodes in the mobile ad-hoc network. Routing protocols for ad-hoc networks have been proposed based on these metrics. Two such proposed routing schemes are: Associativity Based Routing by C.K.Toh [16]and Signal Stability Based Adaptive Routing by Dube [17]. McDonald and Znabi [18] suggest the usage of probability for path availability estimation based on link state information from previous instants of time. Similarly, Jiang [8] suggests the use of path availability prediction wherein the estimation is made and corrected over a period of time starting from the instant a link is formed between the two nodes. As a result, this technique is observed to be more accurate than McDonald and Znabi [18] in the case of short-term path availability estimation.

These techniques are proposed based on the assumption that the networks are not highly mobile. When the nodes are highly mobile, the accuracy of such predictions is questionable.

Flooding vs Clustering

There have been two kinds of approaches to solving the problems in ad-hoc networks. One is the usage of flooding and another is node clustering. The flooding technique has been proposed for exchanging information and locating a destination point under the assumption that the nodal movement is unpredictable in ad-hoc networks. Clustering is based on the belief that there is some kind of pattern or correlation in the movement of the nodes in ad-hoc networks.

The need for flooding is emphasized by scenarios when the life of any link in the path between a source and destination is extremely short due to high node mobility. Flooding helps to discover or reach a destination in the shortest period of time but at the cost of huge bandwidth.

Since bandwidth is scarce in ad-hoc networks, clustering methodologies evolved[19, 20]. Clustering helps in a more focused transmission of information and in efficiently reaching a destination instead of wasting resources or bandwidth by trying all possible paths simultaneously as in flooding. Combinations of flooding and clustering techniques have been widely used in building hierarchy in ad-hoc networks by localizing flooding within clusters while using demand-based destination location among clusters[21].

Clustering has evolved from grouping of nodes in the form of static predefined groups to dynamically discovered groups based on the correlation in mobility[9, 10, 22, 23]. Nodes can be clustered at run time based on their velocity correlation. Wang and Li[6, 7] have used this technique for solving the research problem as in this thesis. Partition prediction completes the clustering mechanism by detecting the movement of nodes away from the cluster.

For quick response and maximum reachability, especially in small ad-hoc networks, flooding is still the technique for information dissemination[24, 25] since the performance of Wang and Li[6, 7] does not measure well when the nodes exhibit significant mobility and random movement.

Ho et al.[25] propose flooding for multicasting in ad-hoc networks, especially for scenarios with high mobility. Viswanath et al.[26] suggest the usage of variations in flooding techniques based on network conditions, as flooding proves to be highly expensive in dense networks. Williams and Camp[27] provide a comprehensive analysis of different flooding techniques proposed so far in ad-hoc networks. The self-pruning technique described by Kim[28] is a combination of flooding and pruning based on the common neighborhood set proposed for multicasting in ad-hoc networks.

No algorithm proposes pruning or optimization of flooding based on the mobility of the nodes. A flooding technique that is information sensitive, i.e., sensitive to the staleness of information due to mobility has not yet been proposed. We propose the usage of service broadcasting in combination with information sensitivity to solve our research problem.

2.1.2 Information Dissemination in Ad-hoc networks

Strategies for high availability of information in ad-hoc networks amidst network partitioning is proposed in [6, 7, 29]. These research works vary in the type of data servers used and the type of data whose availability is in question.

In the paper about information dissemination in partitionable mobile ad-hoc networks[29], Karumanchi et al assume the usage of fixed data servers distributed all over the network whose stored information is variable. Information is queried from or updated to a subset of reachable servers. The assumption here is that the nodes move in small clusters and hence will always have access to atleast one data server within the cluster.

Wang and Li[6, 7] use a class of data servers that can be dynamically replicated and terminated. The stored information is read-only. The clusters are dynamically identified or dynamically formed in these algorithms. Wang and Li propose to optimize the cost of availing information by exploiting the correlation in the movement of nodes, which is highly applicable to an environment where nodes move mostly in groups and the mobility is not high. Having an application layer aware of the mobility helps to provide better service to users[30].

Different types of information dissemination have been discussed in Kulik [31]. They are ideal dissemination(along shortest path routes), flooding, gossiping and SPIN family protocols. The term gossiping refers to forwarding information to a random subset of neighbors of the forwarding node. SPIN family protocols are information sensitive hence perform better than other dissemination protocols.

2.1.3 Mobility Models

In order to thoroughly simulate a new protocol for ad-hoc networks, it is imperative to use a mobility model that accurately represents the mobility pattern of the nodes that will eventually utilize the given protocol. Only in this type of scenario is it possible to determine whether or not the proposed protocol will be useful when implemented. Since there are no trace-based mobility models available, any proposed solution for an ad-hoc network problem has to be evaluated against the available synthetic mobility models to get an estimated measure of its performance under different scenarios caused by mobility.

There are two classes of Mobility Models:

* Random Mobility models and

* Group Mobility models.

Random Mobility models emulate independent movement of nodes in the network, and Group Mobility models emulate movement of nodes in groups.

(1) Random Walk Mobility Model

Random Walk[32] is an emulation of erratic movement. In this model, a node moves with a chosen velocity and travels for a constant period of time or distance. The speed and direction are chosen from a predefined range of [minspeed, maxspeed] and [0, 2π] respectively. If a mobile node reaches the simulation boundary in the course of movement, it bounces off the simulation boundary at an angle determined by the incoming direction.

This model exhibits a memory-less mobility pattern because it retains no knowledge concerning past locations and speed. As a result, it generates unrealistic movements such as sudden stops and sharp turns. If the constant time or distance used for traveling in one direction is small, the network exhibits static behavior.

(2) Random Waypoint Mobility Model

In Random Waypoint Mobility Model[33], a mobile node moves to a new location by changing its velocity and pauses for a random period of time between the changes in velocity. When the pause time elapses, the mobile node chooses a random destination and speed in the range of [minspeed, maxspeed] and travels to that destination. The pause time is chosen from a predefined range of [minpausetime, maxpausetime]. If a mobile node reaches the simulation boundary in the course of movement, it bounces off the simulation boundary at an angle determined by the incoming direction.

In this model, there’s a complex relationship between node speed and pause time. A scenario with fast nodes and long pause-times produces a more stable network than a scenario with slower mobile nodes and shorter pause times. This model generates more or less constant percentage of neighbors for all the nodes in the network.

(3) Modified Random Direction Mobility Model

In Modified Random Direction Mobility Model[34], a mobile node chooses a destination in the chosen direction, travels to that point and pauses for a brief period of time. Direction and speed are varied as in the Random Waypoint model. The mobile nodes are no longer forced to travel till the simulation boundary and hence produce more realistic movements.

This model was designed to overcome the clustering of nodes near the center of the simulation area produced by Random Waypoint Mobility Model. The average hop count for data packets in this model will be much higher than that produced in most other models.

(4) Gauss-Markov Mobility Model

In Gauss-Markov Mobility Model[11, 35], the randomness of nodal movement is varied using a tuning parameter. Pure random movement is obtained for a tuning parameter equal to 0 and linear movement for a tuning parameter equal to 1. By definition, the movement of a node undergoes a change in its velocity based on its movement in the previous fixed interval of time n.

Change in speed s_n and direction d_n, at the nth instance, is based on the following equations.

$\mathrm{s_n} = \alpha s_{n-1}$+(1-$\alpha)\overline{s} + \sqrt{(1-(\alpha * \alpha)}s_x$

$\mathrm{d_n} = \alpha d_{\mathrm{n}-1}$+(1-$\alpha)\overline{d} + \sqrt{(1-(\alpha * \alpha)}d_x$

where α is the tuning parameter, $s_{\mathrm{n}-1}$ and $d_{\mathrm{n}-1}$ are the speed and direction at (n-1)th instant, $\overline{s}$ is the mean speed and $\overline{d}$ is the mean direction computed over all the previous time instants, s_x and d_x are random variables from a Gaussian distribution.

During the time instant k in the time period interval n, the node's movement in two dimensional space ($\mathrm{x_k}$, $\mathrm{y_k}$) at instant k is given by

$\mathrm{x_k}=\mathrm{x}_{\mathrm{k}-1}+\mathrm{s}_{k-1}\mathrm{cosd}_{k-1}$

$\mathrm{y_k}=\mathrm{y}_{\mathrm{k}-1}+\mathrm{s}_{k-1}\mathrm{sind}_{k-1}$

where ($\mathrm{x}_{\mathrm{k}-1}$, $\mathrm{y}_{\mathrm{k}-1}$) is the position of the node in two dimensional space at time instant k-1.

Since this model moves based on the speed in the previous instant, it produces more realistic movements by eliminating the sudden stops and sharp turns encountered in Random Walk Mobility Model.

(5) Reference Point Group Mobility Model(RPGM)

Reference Point Group Mobility Model[6, 36] represents random motion of a group of nodes and random motion of nodes within the group. Group movements are based on the movement of a group reference point. Based on the group reference point, other individual nodes define their location with small displacements from the group reference point. As a result, the motion of the group reference point defines the direction of motion of the nodes in the group.

With appropriately chosen parameters, this model can emulate all possible group mobile application scenarios. The only drawback in this model is the lack of dynamic group membership. Nodes follow the same predefined group. The RPGM model is explained in detail by Lin and Gerla [22], and a mathematical expression is derived by Baochun Li[6].

(6) Reference Velocity Group Mobility Model

In Reference Velocity Group Mobility Model[37], the members of the group follow a group velocity with slight deviation. The deviation is a random velocity vector of smaller magnitude than the group velocity. The members of the group emulated in this model exhibit more correlated movement in the velocity space rather than movement in clusters in displacement space.

The advantage of this model is that the group velocity and individual deviation vector can arbitrarily follow any model to simulate various mobility patterns that may exist for different groups and nodes within those groups.

The drawback in this model is the lack of dynamic group membership. The model used in Wang[7] provides for dynamic group change by allowing a percentage of the nodes to follow the random walk model. As a result, these nodes that exhibit random motion undergo change in group membership at runtime.

A good analysis of the existing mobility models is given in Camp and Davies[15].

2.2 Related Work

There are two algorithms proposed for the research problem we explore. These algorithms address the issue from the point that the ad-hoc network nodes move in groups and rarely exhibit random or independent movement. We will briefly explain the two algorithms in this section.

(1) Distance Based Clustering Algorithm

An adaptive distributed algorithm for maximizing service efficiency is proposed by Li[6] to provide QoS in ad-hoc networks. This algorithm is distributed in order to reduce the effects of a highly dynamic topology while incurring minimal message exchange overhead. The algorithm is strictly based on group movement where nodes do not change group membership at run time. This algorithm enables an ad-hoc network node to identify its group members at run time.

Each node constructs a list of long-term neighbors by sampling distance with nodes within its transmission range over a period of time and eliminates a node from its list if it is not found in its vicinity for a threshold period of time. Each node then exchanges membership information with the identified neighbors. As a result, all the nodes have knowledge of the nodes in its group.

The number of service instances in a group is kept at a minimum value of 1 by least-id leader election. Whenever a node moves out of the transmission range of its group members, it tries to replicate service accessible through a non-group neighboring node if any. The loss of service is considered transient as the node is assumed to join its group after a period of time. If a temporarily isolated node hosting a service instance joins its group again, the least-id service instance is chosen to be the service provider and other service instances are terminated.

Service Efficiency is defined as a ratio of number of nodes reachable by a service instance to the number of service instances hosted in the network. Since the number of service instances in a group is kept at minimum, the service efficiency is high. The algorithm incurs 8 to 12 service instances for a network of 100 ad-hoc mobile nodes with a transmission range of 60m in a simulation area of 800m*800 m[6].

There are two drawbacks in this algorithm:

1. When a node strays away from its group, it is at a loss of service until it has some neighboring node that has access to a service provider. When a small group instead of a node strays from the group with a service instance, the whole group suffers from service interruption until they encounter a node through which they can access a service instance.

2. If a node changes its group, it replicates a service instance through one of those group members until both the group and itself identifies its membership in the group. It hosts the service instance for time equivalent to the group membership determination threshold.

(2) Velocity Based Clustering Algorithm

To overcome the service interruption due to unpredicted node isolation, Wang and Li[7] used velocity correlation to identify stable groups and predict network partitioning. The service instance collects velocity information from the clients as piggyback information along with each client's service request. This algorithm uses a Kalman filter to detect node clusters in velocity space. Thus, when a client moves away from the service instance, the service is able to detect isolation of the client node.

When the service instance senses the client node's departure, the service instance sends a replication request to the client. The client accepts or declines the server request based on the availability and reliability of alternate service instances if any. A client chooses its service provider based on a reliability counter of how long the service is in the group of the client.

If multiple services exist in a group, the services monitor each other's presence and choose the highest-id server with reliability counter exceeding the turnoff threshold as the service provider of the group. Based on the combination of these strategies, the algorithm guarantees service accessibility by spawning a new service when network partitioning is predicted.

The drawback of this algorithm is that it cannot guarantee service coverage in a highly dynamic environment since reliability counter thresholds used for service replication, service termination and choice of service provider places a limitation on the adaptability of the algorithm. When all the nodes in the ad-hoc network do not exhibit correlated movement, the service coverage drops to between 80% and 90%, whereas the service cost varies from 0.75 to 2.0 (in terms of the number of mobility groups)[7].

One main assumption in the above work that is reasonable but not always an occurrence is that the nodes move in small teams and the communication is within the team in order to coordinate movements. As a result, the algorithm always looks for correlation in the mobility pattern of nodes, and when the nodal movement is random, it assumes the entire network to be part of one group. As a result, the number of service instances in the network goes below the required number which is equivalent to the number of connected components in the Graph formed by the nodes in the network.

Non-Clustering Approach

The goal of this thesis is to provide uninterrupted and continuous access to services in ad-hoc networks amidst network partitioning. Though the objective of this study is the same as the above work(s), we differ in our approach in that we do not attempt to gather any clustering information and use controlled message-passing techniques in combination with some simple prediction rules for QoS provisioning. We do not assume that nodes move in groups.

We provide for information about service instances at all nodes in the network so that when a node wants to access the service, it can make an immediate decision and access a service without performing service discovery.

Gathering of clustering information will prove to be more useful if the network is stable to some extent. Making use of clustering information makes the algorithm to be less responsive to quick changes in the group membership.

We will not attempt to provide fixed guarantees on the QoS offered as the connectivity of the network, bandwidth and link availability are variable in ad-hoc networks. This inherent stochastic communications quality in a wireless ad-hoc network makes it difficult to offer fixed guarantees on the services offered to a device. In networks of this kind, fixed guarantees would result in requirements on how nodes move as well as requirements for node density, which would inherently inhibit the notion of ad-hoc operation[44].

In short, we will illustrate that the service coverage can be guaranteed without using any clustering approach in ad-hoc networks.

2.3 Problem Definition

2.3.1 Assumptions

The system consists of a set of n independent mobile nodes communicating by message-passing over a wireless network. The network is modeled as highly dynamic with the least possibility of stable grouping of nodes for a significant period of time. Assumptions on the mobile nodes and network are:

- All nodes have equal capabilities and are capable of hosting services if required.
- All communication links are bi-directional, i.e., all nodes have same transmission range.
- The services are adaptive, i.e., replicate-able, and can be created and terminated dynamically.
- Network propagation time between any two neighboring nodes is insignificant.
- Nodes communicate without interference and collision.
- Downloading time for the service hosting information is negligible.

- All nodes in the network use the same algorithm and strictly follow the protocol for QoS provisioning.
- A link level protocol, that ensures that a node is aware of the nodes within its transmission range, is assumed.
- Downloaded information is time sensitive.
- Services cannot be statically provisioned and deployed as a fixed infrastructure.

No assumption is made about node identification semantics.

2.3.2 Problem Statement

All n nodes in the ad-hoc network should have access to at least one service instance while keeping the number of service instances(service cost) fewer than the total number of nodes in the network.

The lower bound of the service cost is defined to be equal to the number of connected components in the Graph G formed by the nodes in the ad-hoc network where the edges of the Graph corresponds to the existence of communication link between adjacent pairs of nodes. In a connected component of a Graph of nodes, there exists a path between every pair of nodes. As a result, if we place a service instance in each of the connected component, service coverage can be provided for all the nodes in the ad-hoc network. Due to mobility, composition and the number of connected components change continually. As a result, the average service cost will be higher than the lower bound inorder to provide for continuous service coverage in ad-hoc networks.

In order to provide a complete solution, the algorithm should address the following aspects of the problem:

- **Distributed and localized decision making.** Ad-hoc networks emphasize the need for peer-to-peer architecture due to administrative independence of the nodes. Relying on a centralized service to provide for QoS provisioning is rendered impractical due to the following reasons: (1) unreliability of path availability and (2) accessibility of centralized service is a node-specific problem hence can be best solved by the node itself.

- **Minimization of Service Cost**, i.e., minimizing the number of service providers at any instant in the network.
- **Sufficiently long access to a service provider** inorder to reduce the overhead of switching service providers.
- **Reduction in the need for migrating service providers** from one hosting node to another in order to lower the cost of downloading and bringing up the service instance.
- **Minimal message exchange overhead** in order to reduce the adverse effects of high mobility and limited bandwidth in wireless networks.

The number of service instances required for providing the necessary coverage depends on the density of the network, degree of mobility of the nodes and connectivity of each individual group in the network.

2.4 Proposed Approach

The non-clustering based algorithm depends on the service provider information available to the nodes in the ad-hoc network. Based on the service provider information, the algorithm, run on each node, decides to access or host a service provider. For quick notification of service provider information, we use outward information flow mechanism which notifies all the service provider on the order of the depth of the spanning tree routed at the service provider. The service provider which has least hop reachability and accessible through maximum number of neighboring nodes is accessed. This is an approximation of the k-shortest path[45, 46, 47, 48] and disjoint path finding algorithm[49, 50] problems. The k-shortest path algorithms are of the order of nlogn+m where n is the number of nodes and m the number of edges in the Graph formed by the nodes in the network. In the k-shortest path algorithms, a path checking rule is applied to check conformation to constraints. We do not perform this phase as the information is valid only for a short duration due to mobility of the network. Some optimization heuristics are applied to this strategy to eliminate common communication problems like looping. We then propose a simple QoS provisioning algorithm that, based on the information available to the node at that instant, does the following:

(1) chooses the most suitable service provider,

(2) if isolation is predicted, downloads the service hosting information,

(3) when no node is accessible, hosts a service instance in the node itself, and

(4) brings down the service instance when it is no longer required.

Heuristics applied to achieve the above is given in detail in Section 3.1.1.1. Since the nodes are mobile, collecting information about all reachable nodes might be time consuming and hence not very useful. Time delay in gathering information about reachability of nodes implies stale information.

Though the message passing mechanism can be considered as a special case of flooding, it differs from the basic flooding technique in that the decision to forward the information is made at each node before further propagation. The decision is mobility sensitive. The performance of this message passing technique is expected to be better than the flooding technique as it uses information sensitive pruning techniques. We exploit the redundancy in information exchanged for identifying a highly accessible and stable service provider.

2.5 Conclusion

Literature survey in the field of wireless ad-hoc networks revealed that the non-clustering approach is unexplored for the problem of providing uninterrupted and continuous service coverage in adaptive services-aware ad-hoc networks. Surveys of mobility models suggest the usage of combination of mobility models to simulate realistic mobile environment scenarios. We then enumerated the requirements on the performance of the algorithm for it to be a realistic solution. We propose a simplified QoS provisioning algorithm based on a simple message passing technique as a possible solution for the above-defined problem.

In the next chapter, we will discuss in detail the proposed algorithm along with the optimization heuristics.

Chapter 3

Algorithm for Adaptive Service Provisioning

In this chapter, we briefly outline our strategy and algorithm for adaptive service provisioning. Then, we discuss in detail the different aspects of the proposed strategy and the rationale behind the different schemes used to optimize the performance of the strategy.

There are three aspects to the proposed strategy. First is the information exchanged by the nodes in the network; second is the processing associated with the propagated information, and third is the QoS provisioning algorithm run locally at each node. No extrapolation of information gathered over a period of time is made for the simple reason that the information grows stale due to the mobility of the nodes. The basic idea of the algorithm is that nodes exchange information about the service providers accessible through them periodically and do not need to know about the reachability of non-service-hosting nodes.

3.1 The Algorithm

The proposed algorithm is an augmentation to the general information propagation technique or the wave algorithm used in distributed systems. In this technique, information received from one node is transmitted to all other neighboring nodes except the node from which it received the information. As a result, the information propagates outward like a wave from the source. There are three phases to the algorithm that are run at periodic intervals. Here we briefly outline the phases.

(1) **Information Exchange/Propagation:**

Node propagates service provider related information gathered in the previous instant to its current neighbors. The information received from ex-neighboring nodes is pruned before transmission. The service provider and the minimum number of hops necessary to reach the provider through the node are propagated.

(2) **Information gathering:**

The node processes information, received from its current neighbors, to identify the accessibility and connectivity of service providers. This helps the QoS provisioning phase to identify the service providers that can be accessed for a longer period of time without interruption.

(3) **QoS provisioning:**

This is more of a decision making phase. The decision to access a service provider and to create a service provider, when necessary, is the essence of this phase.

If the current service provider of the node is not accessible, the node starts accessing the service instance that is predicted to be the most accessible service provider. Information necessary to host a service is downloaded if isolation is predicted. A service instance is hosted with downloaded information when there is no longer any accessible service instance. The check for redundancy of the hosted service is made if the node is a service provider.

Initially when the node does not have access to a service provider, it cannot avoid the situation of being without service.

3.1.1 Information Exchange and the associated processing

Nodes in the network exchange service provider information available to them. At any node, the information gathered at that instant forms the basis for information exchange and QoS provisioning in the next instant of time. To start with, all the nodes hosting service instances in the network inform their neighbors that they are service providers. These neighbors in turn inform their other immediate neighbors that such and such service provider is accessible through them and is one hop away. This information gets propagated in a wave-like fashion to all the nodes in the network as shown in Figure 3.1. In Figure 3.1, the black dotted lines indicate the existence of a link between the connected

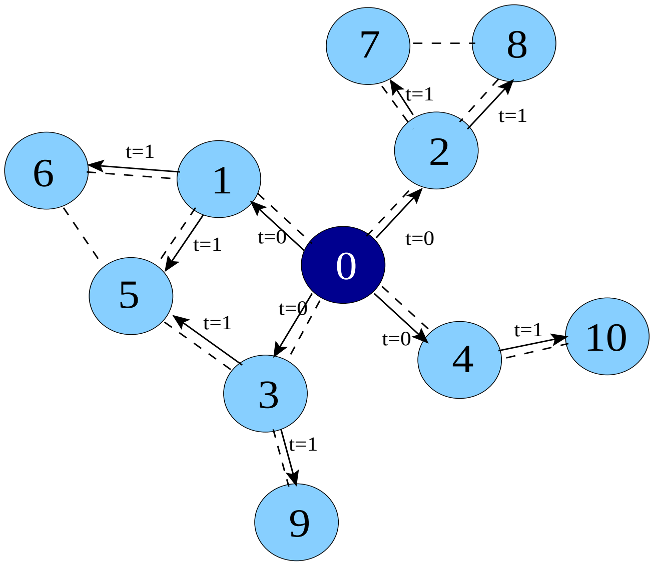


Figure 3.1: Information Propagation in a wave -like fashion originating from the service provider

nodes. This is similar to the information broadcasting technique of distributed systems. Information is processed before forwarding to other neighbors in order to refine the above technique and optimize the information flow.

We use three data structures to facilitate efficient information propagation and simple QoS provisioning. They are Previous Instant List or Previous Instant Information List(PIIL), Current Instant List or Current Instant Information List(CIIL) and Accessibility List(AL). We use the term Reception Interval(RI) to refer to the time period required by a node to send the service provider information to all its neighbors or the time taken to receive information about at least one service provider, whichever is larger.

We use the two lists, Accessibility List and Current Instant List, for storing information as it is received from neighbors. The Accessibility List is used to keep track of the number of neighbors through which each service provider is accessible. Current Instant List is the list that is currently being updated with the information received. Previous Instant List is the list of information received during the previous reception interval. At the end of one Reception Interval, Current Instant List is copied into Previous Instant List and the Accessibility List is again built along with Current Instant List. Information propagation and QoS provisioning is based on the information that was gathered in the previous instant and the knowledge of current neighbors.

Our experimental implementation had these two lists implemented as a list storing the tuple < *Service Provider, the neighbor through which it's accessible, minimum number of hops required to access the Service Provider through this neighbor*> sorted in the increasing order of hop count. The Accessibility List stores the list of Service Providers sorted on the number of immediate neighbors through which it is accessible. When a message arrives from a neighbor, the Current Instant List and the Accessibility List are updated. The information stored in the previous instant list is propagated to the neighboring nodes every time the current instant list is copied into the previous instant list. Only minimum-hop entries corresponding to each service provider in the previous instant list are sent to the neighbors.

3.1.1.1 Optimization Heuristics

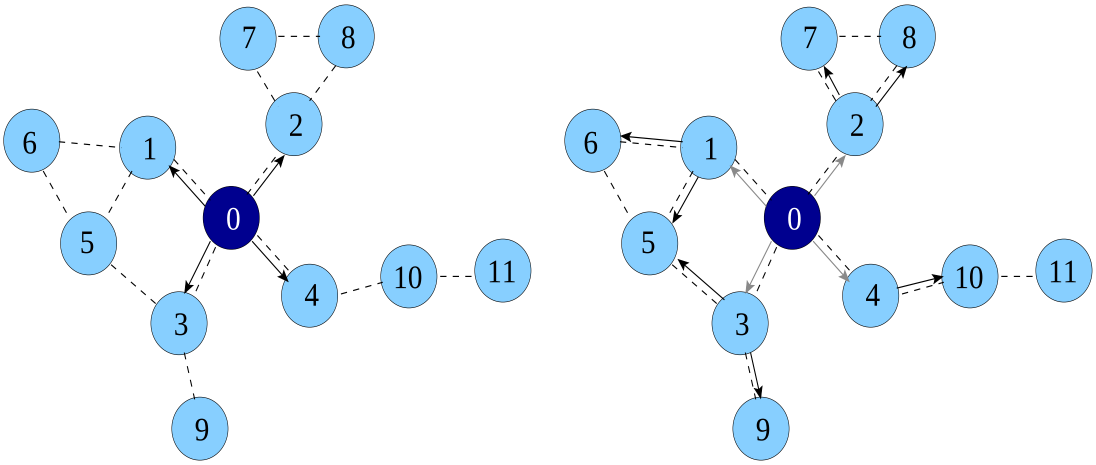
Controlled flooding technique is used for information propagation, as the topology of the network is uncertain at any instant. Due to mobility, information exchanged gets obsolete soon. To avoid stale information from being propagated, the node discards the service provider information gathered in the previous instant from ex-neighboring nodes. Thus, we eliminate the counting-to-infinity problem in the case of such stale information.

The higher the mobility, the shorter is the life of the information. Looping of information is prevented by (1) not sending the information obtained from a neighbor back to the neighbor (direct looping), (2) not sending information about a service provider to the service provider itself (indirect looping). Sending only the service provider information with the minimum number of hops helps to keep the message succinct and prevents looping.

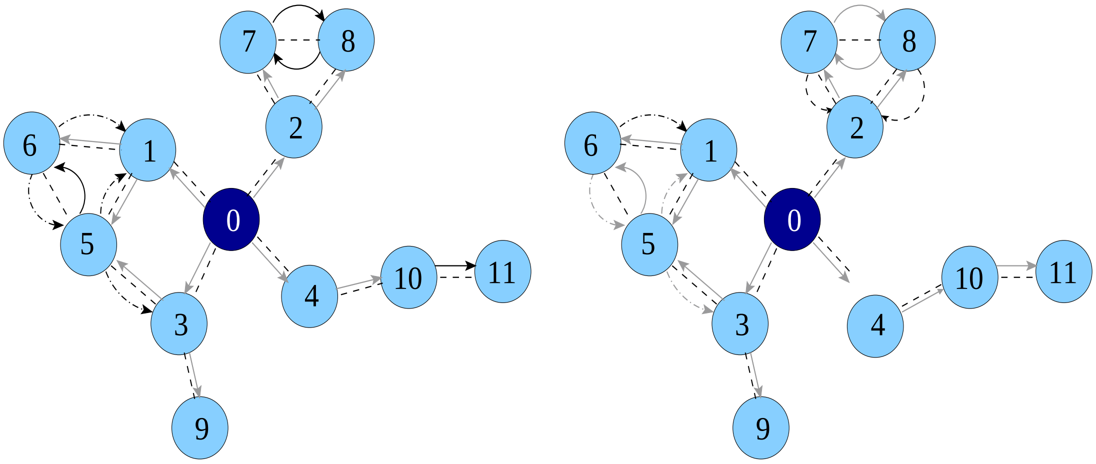
Using Figure 3.2, we will illustrate how information propagation based on hop count minimizes the propagation of redundant information and avoids looping.

The resultant flow of information about a service provider, generated at any instant, is illustrated in Figure 3.2. The dotted lines indicate the existence of a link between the concerned nodes.

In Figure 3.2(a), the service provider (node 0) propagates to its immediate neighbors that it's a service provider as <Service Provider=0, hops=0>, where hops indicate the number of hops it takes to reach the Service Provider through node 0. Nodes 1,2,3 and 4 receive the information propagated at time instant t=0. Since this piece of information received from node 0 has the fewest number of hops,



(a) At time instant t=0 (b) at time instant t=1



(c) at time instant t=2 (d) at time instant t=3

Figure 3.2: Propagation of Service Provider Information over a period of time

these nodes propagate it to their immediate nodes in the next instant. Since direct looping is avoided and the service provider information is not sent to the service provider, the message propagates outward from the original source of the message, i.e., the service provider.

At time instant t=1(refer Figure 3.2(b)), node 10 receives the modified message <Service Provider=0, neighbor=4, hops=1> from node 4. Node 5 and Node 9 receive the modified message <Service Provider=0, neighbor=3, hops=1> from Node 3. Node 5 and Node 6 receive <Service Provider=0, neighbor=1, hops=1> from Node1. Node 7 and Node 8 receive <Service Provider=0, neighbor=2, hops=1> from Node 2. It can be observed that Node 5 will be able to access the Service Provider through two nodes with the same number of hops. When each node propagates information about any service provider, it always sends only one message to its neighbors with the minimum number of hops required to access the service provider through it. So the messages from Node 1 and Node 3 will be consolidated by node 5.

In the next instant, Node 3 will receive information about the service provider from Node 5 due to the message received from Node 1(refer figure 3.2 (b)). Though this propagation appears like looping, it gives an idea of the alternate paths available to the node to access the same service provider. All propagations that occur at time instant t=2(refer Figure 3.2(c)), in dash-dotted lines do not trigger generation of new modified messages by the receiving nodes for the simple reason that their hop count is more than the minimum number of hops, in which the service provider can be accessed from the receiving node. They add to the knowledge of the accessibility list. Accessibility list stores the number of neighboring nodes through which a service provider is accessible to the node. The solid lines indicate the message propagation, which results in further propagation in the following instants of time. So the information about the service provider effectively reaches all the nodes in a time equivalent to the depth of the spanning tree rooted at the service provider.

During the time interval between the propagation of a message by Node 0 and the decision to propagate modified version of the message by its neighbor 4, if the Node 0 goes out of the transmission range of Node 4, the message is not propagated. Here pruning for staleness helps to avoid the counting-to-infinity problem. Consider a scenario as illustrated in Figure 3.2(d), the loss of link between Node 0 and Node 4 is not known to node 10, and node 10 goes on to propagate the information to Node 11. To minimize the effect of this problem, we resort to accessing a service provider that can be accessed through the maximum number of immediate neighbors. This problem cannot be fully resolved, as the change in the connectivity beyond one hop is not known at the same instant of time.

3.1.2 QoS Provisioning

Given the above framework, QoS provisioning is more of a decision making process than an algorithmic procedure.

The key to accomplishing the objectives of the algorithm lies in:

- **Choice of Service Provider:** The service provider that can be accessed through maximum number of immediate neighbors in that instant of reception interval is chosen to be the service provider. This is because more alternate paths are available to access the service provider. So there is a high chance to access this service provider for a longer period of time. The choice of service provider also minimizes the effect of inaccurate or stale information, if any, about service providers.

- **Rule for Service Termination:** When all the neighboring nodes have access to other service providers and the service provider has not been isolated in the previous instant, then the service instance is brought down. The checking for node isolation in the previous instance is to ensure that the node is not transient in the neighborhood of new nodes. This is to prevent the cost of bringing down the service instance just to bring it up again in the next instant.

- **Isolation Prediction:** When only one service provider is accessible and its accessibility falls below an availability threshold, it is inferred that the node is getting isolated. This inference may not be accurate when the service provider and the node have correlated movement. Inaccurate prediction results only in downloading the information.

- **Rule for hosting Service Instance:** This is trivially accomplished when the node cannot access a service provider and has already downloaded service information. When isolation is anticipated, nodes download information and host only when required thus keeping the number of service instances low in the network.

To accomplish the above, the only information required to exchange is the list of service providers accessible through "which" neighbor and the minimum number of hops required to reach the service provider.

3.2 Formal Description of the Algorithm

3.2.1 Notation

Let N denote the total number of nodes in the network.

Let *RI* denote Reception Interval.

Let *Neighbors(a)* denote the list of neighbors of node *a*.

Let *Previous_Instant_List(a)* and *Current_Instant_List(a)* denote the list of Service Instances together with minimum number of hops and the neighbor of ***Node a*** through which it is accessible. It has multiple entries in terms of hops or service provider. i.e., service instances can be reached through multiple neighbors.

Let *Current_Instant_List(a)* denote the list being built in the current instant of Reception Interval.

Let *Previous_Instant_List(a)* denote the *Current_Instant_List(a)* of the previous instant of RI less the information received from ex-neighboring nodes.

Let *AccessibilityList(a)* denote the list of service providers and the number of neighboring nodes through which they are accessible.

Let *PropagationList[*a,b*]* denote the list of service instances accessible through *node a* together with minimum number of hops required to reach the service instance. It does not include the information sent by *node b* to *node a* in the previous instant, if any. It does not contain any information sent by $c \notin \text{Neighbors}(a)$ in the current instant of time.

Let *Redundant_Service_Instance_Rule* refer to the rule used for service termination.

Let *Current_Service_Provider of a node* refer to the service provider that is currently accessed by the node.

Let *Isolation_Prediction_Rule* refer to the rule used for Isolation Prediction.

3.2.2 Description

Information Propagation

At Node *a*:

```
During the Reception Interval RI,
    for each $b \in Neighbors(a)$
        Build PropagationList[a,b]
        Send PropagationList[a,b] to node b
    end for
```

Information Processing

At Node a:

```
During the Reception Interval RI,
    Receive PropagationList[b,a] where $b \in Neighbors(a)$
    foreach entry <hops, ServiceInstance> in PropagationList[b,a]
        Increment AccessibilityList entry for ServiceInstance
        Insert an entry <hops, ServiceInstance, b> in Current_Instant_List[a]
    end for
At the start of the Reception Interval RI,
    Copy Current_Instant_List(a) to Previous_Instant_list(a)
    Perform QoS Provisioning
    Clear Accessibility_List(a)
```

QoS provisioning

At Node a:

```
    if Isolation_Prediction_Rule== true
        Download Service Hosting Information
    end if
    if Current_Service_Provider of Node a is not accessible,
        Access the Service Instance with maximum count in AccessibilityList[a]
        if AccessibilityList[a] is empty
            Host Service Instance with downloaded information if any
        end if
```

```
endif
if Node a is running a ServiceInstance,
  if Redundant_Service_Instance_Rule==true
    Shutdown the ServiceInstance running in Node a
  end if
end if
```

3.3 Algorithm Complexity

3.3.1 Information Propagation

Since the bandwidth of a wireless channel is limited, an analysis of the bit complexity and time complexity of the propagation of a single piece of information through the network is warranted.

3.3.1.1 Communication Complexity

Message Complexity is defined by the number of messages used for propagation of information about a single service instance across the network. One message corresponds to an entry in the Propagation-List sent from Node *a* to Node *b* (Refer 3.2.1).

Lemma. 1. The message complexity of information about a single service instance in the ad-hoc network is of the order of number of edges in the set of nodes reachable from the service instance.

Proof: To simplify our analysis, let us assume the network to be static during the period of propagation of a single information. Then the network can be considered as an undirected graph G .

Let the number of nodes in the graph be N.

Let e denote the number of links/edges connecting the nodes in G. Each edge connects two nodes which are within the transmission range of each other.

Message complexity = number of edges in the Graph G due to outward flow from the service instance + number of edges in Graph G due to message exchange about alternate paths - degree of the node hosting service instance (no direct looping)

= O(sum of the degree of nodes in the graph) - degree of the node hosting service instance= $O(e)$ where e is the number of edges in G.

Now let us address the bit complexity of a message.

Lemma.2. The bit complexity of the message (single information) is $O(\log n)$).

Proof: The information that is propagated is the identifier of the service provider, identifier of the sending node (along an edge) and the minimum number of hops it takes to reach the service provider through the sending node.

If x is the number of bits required for identification of nodes in the network and n is the number of nodes in the network, then the number of bits it takes to construct a message is $2x+\log n$.

If $x<\log n$, then the bit complexity is $O(x)$ else $O(n)$. For example, if the node identification mechanism is as simple as numbering nodes from 1 to n (number of nodes in the network), then the bit complexity is ($2\log n + \log n$) i.e., $O(\log n)$).

Lemma.3. The maximum size of the composite message sent from one node to another is $O(n\log n)$.

Proof: Each node sends a message information to another node about each service provider.

If p is the number of service providers, then the node sends at most $p*3\log n$ messages.

Worst case scenario for p will be $(n-1)$ where all nodes in the network are service providers.

Then the bit complexity of composite message is $(n-1)*3\log n$, i.e., $O(n\log n)$).

The worst-case scenario will occur only when information flow about the service provider reaches all the nodes over a period of time due to mobility and loss of connectivity in path due to mobility is undetected and all the nodes host service instance at one time or the other due to high mobility before this composite message propagation occurs at a particular node. This scenario is rare but a possible occurrence.

3.3.1.2 Time Complexity

Time Complexity is defined by the number of time instants it takes to propagate a message to all the nodes in G.

Lemma. 4. The Time Complexity of information propagation is $O(n)$.

Proof: Since the algorithm functions like a wave, the time complexity is given by the length of the longest chain of nodes in the spanning tree rooted at the source n0, ie., worst case possibility of the length of the spanning tree (all nodes constituting the chain) which is nothing but *O(n)*.

3.3.2 Information Gathering

3.3.2.1 Computational Complexity

The calculation of Computational Complexity is shown with respect to one node in the group. Here we split information processing into two : (1) Information Reception and (2) Information Propagation.

(1) Processing associated with receiving propagation list from Neighbors

Let *Ng* be the set of nodes in a group G and S_g be the set of Service Instances in the group G. Let's assume the worst case possibility where all nodes b ∈ *Neighbors(a)* of Node a in G sends information to Node a about all s$\in S_g$. Then,
Receive PropagationList is executed |*Neighbors(a)*| times.
Insertion into Current_Instant_List[*a*] for each reception is $|S_g|$ insertions.
Assuming the usage of a combination of hash-table of service instances and binary sorted list <*hop, neighbor*> for that service provider,
Cost for one insertion =cost for finding the sorted list of a service provider + cost of insertion in a binary sorted list

$= O(1)+O(\log|Neighbors(a)|)$.

Number of insertions into Current_Instant_List(a) per reception interval RI, $n_i = |Neighbors(a)| * |S_g|$ insertions.
Amortized Cost for n_i insertions

$= {}_{k=1}{}^{to}\text{Neighbors}_{(a)}\sum \log k + |S_g| * (\sum O(1))$

$= \sum \log(|Neighbors(a)|!) + O(|S_g|)$

$= \sum \log(|Neighbors(a)|)^{|Neighbors(a)|} + O(|S_g|)$ using Sterling's formula

$= O(|Neighbors(a)|.\log|Neighbors(a)|) + O(|S_g|)$.

(2) Information Propagation

Information Propagation results in Build Propagation List executed |*Neighbors(a)*| times.

The Propagation List is constructed |*Neighbors(a)*| times. The components of the Propagation List are the first or the second entry of the sorted list of each service provider.

The second entry in each list is accessed only once for every set of *Neighbors(a)*.

So the Amortized cost for BuildPropagationList for |*Neighbors(a)*|

$= (|Neighbors(a)-1|*O(1)+O(2)*|S_g|$

$=O(|Neighbors(a)|)+O(|S_g|)$.

So the Computational Complexity associated with Information Exchange with respect to one node when all its neighbors transmit information about all service instances in the group

$=O(|Neighbors(a)|.\log|Neighbors(a)|)+O(|S_g|)+O(1)*(|Neighbors(a)-1|+O(2)*|S_g|$

$=O(|Neighbors(a)|.\log|Neighbors(a)|)+O(|S_g|)$.

Assume all nodes in the group are neighbors of each other(highly connected graph), then Computational complexity = $O(|N\text{g}|*\log|N_g|)+O(|S_g|)$.

But when the nodes are highly connected, $|S_g|$ tends to 1. Then, Computational complexity tends to $O(|N_g|*\log|N_g|)$.

3.3.3 QoS Provisioning

3.3.3.1 Time Complexity

Since the QoS Provisioning phase is a series of decisions, the time complexity of this phase is O(1) since all decisions are on the order of 1.

3.4 Conclusion

In this chapter, we discussed the algorithm in detail along with its optimization heuristics. We then proved theoretically that the message complexity of the algorithm is O($N_g{}^2$) and the worst case computational complexity is $O(N\log N)$ where N is the number of nodes in the network and all nodes are in the immediate neighborhood of each other.

In the next chapter, we define the metrics and measure the performance of the distributed algorithm in terms of these metrics through simulation.

Chapter 4

Experimental Validation through Simulation

In this chapter, we discuss the different performance evaluation parameters and simulation parameters that are used for the performance evaluation of the algorithm. Then, we discuss the details of the simulation environment and evaluate the performance of the algorithm through simulation against different mobility patterns in terms of the performance evaluation metrics.

4.1 Metrics

Here, we define two types of metrics.

- Performance evaluation parameters and
- Simulation parameters.

Performance evaluation parameters are variables measured to estimate the stability and the performance of the algorithm. Simulation parameters are variables that are varied to generate different scenarios for the simulation.

4.1.1 Performance Evaluation Parameters

We define a few performance evaluation parameters as a measure of the objectives of the algorithm. We use the definition of Service Coverage and Service Cost as defined by Baochun Li and Karen Wang[6, 7].

Service Coverage

Service Coverage is defined as the ratio of the total number of nodes covered by the services to the total number of nodes in the ad hoc network. We extend this definition to encompass two measures: Service coverage from service instance point of view and service coverage from client node point of view. From the service instance point of view, we estimate the total number of nodes reachable by all the service instances in the network. We define service coverage with respect to the client as the total number of nodes in the network that have identified a valid service provider. By valid service provider, we mean a node hosting a service instance and reachable from the client in the current instant of time.

Service Cost

Service Cost is defined as the ratio of the number of nodes hosting the service to the total number of connected components in the Graph G formed by the nodes in the ad-hoc network. A theoretical definition of the lower bound of Service Cost is provided by Baochun Li[6] as the number of connected components in the Graph G formed by the nodes existing at that time instant in the network.

Frequency of change in the Service Provider accessed by a Node

Switching from one service provider to another will incur some overhead. Switching to another service provider should be done only when the current service provider is no longer accessible. Accessing a more stable service instance or a service instance that has correlated movement with the client reduces the frequency of change in the service provider accessed.

Mathematically, if k is the number of instants of time during which a node has a particular service instance as its chosen service provider, then the Frequency of change in the Service Provider accessed is given by the inverse of k.

4.1.2 Simulation Parameters

The proposed strategy is evaluated against different scenarios characterized by variation in mobility pattern and density of the network. Varying the values for the simulation parameters generates different mobility patterns.

Network Size and Simulation Area

Network size denotes the number of nodes in the network. Network size in combination with fixed simulation area provides a good measure of the density of network. The algorithm's performance can be evaluated in the cases where nodes are sparse or dense. A decrease in the number of nodes in a fixed area implies a decrease in the connectivity of nodes i.e., each node has fewer neighbors. A decrease in connectivity also implies lesser information exchange hence less input to the algorithm. An increase in the number of nodes implies high connectivity among nodes; more information is exchanged and hence more input to the algorithm. Connectivity of the nodes in the network can also be varied by varying the transmission range of the nodes.

Simulation Time

If the simulation time is long enough, we can check whether the performance of the algorithm degenerates over a period of time.

Mobility Models

We have chosen the mobility models for our simulation to encompass the three possible types of scenarios where:

(1) all nodes in the ad-hoc network exhibit independent movement,

(2) all nodes in the ad-hoc network exhibit correlated movement with one group of nodes or the other, and

(3) a percentage of the nodes in the ad-hoc network exhibit independent movement while the rest of the nodes move in groups.

(1) Modified Random Direction Bounded/Boundless Model

This model emulates the scenario where all nodes exhibit independent movement. We have chosen Modified Random Direction Mobility Model with variations in order to simulate realistic movements.

This model has two variations used in combination with two types of simulation area. In one variation, we use fixed pause-times between changes in direction, speed and travel time similar to Modified Random Direction Mobility Model[34]. The pause-time is fixed in order to simulate scenarios where all nodes in the network perform the same constant time-consuming operation.

Another variation of this model uses a variable pause-time. The nodes travel with variable travel-time, speed and direction. Bounded or boundless refers to the type of simulation area used. In the bounded model, when the node reaches the simulation boundary in the course of its movement, its direction changes from Θ to $(\prod - \Theta)$. In boundless model[15], the simulation area is modeled in the form of a torus with the vertical and horizontal edges of the rectangular simulation area joined together. Pause-times are kept sufficiently small to avoid the network becoming stable or static. We choose travel time and not a destination, like the Random Waypoint model, to prevent “density waves” near the center of the simulation area.

(2) Reference Point Group Mobility Model

This model emulates only group movement and random movement of nodes within the group. In the Reference Point Group Mobility Model, the nodes in the network are organized into mobility groups. Each group has a logical group center, the reference point, which defines the movement of the entire group. The individual displacement of the nodes in the group follows a random walk of a smaller magnitude compared to the displacement of the group reference point.

(3) Hybrid Reference Velocity Group Mobility Model(Hybrid RVGM)

In this model, we emulate the scenario where a percentage of nodes move in groups and the rest of the nodes exhibit independent movement. The group movement is emulated using the RVGM model and the independent movement is emulated using the Random Walk Mobility Model.

The basic Reference Velocity Group Mobility Model is an extension of the Reference Point Group Mobility Model. In this model, each mobile node is represented by its velocity $v=(v_x, v_y)^T$ where v_x

and v_y are the velocity components in the x and y directions, respectively. Each mobility group has a characteristic mean group velocity from which the velocity of nodes in the group slightly deviates. The node velocity distribution in each mobility group is modeled by a Gaussian distribution parameterized by the mean group velocity and a tuning factor that represents the amount of variation that can exist in the member node's velocity.

In this model, we simulate four cases: 100%, 75%, 50% and 25% RVGM model.

4.2 Simulation Framework

The Simulation framework used to validate and evaluate the performance of our strategy simulates the movement of nodes in mobile ad-hoc networks based on the simulation parameters defined above. All simulation parameters are configurable. Since the problem we try to solve here is an application layer problem, we do not model the lower layers of the protocol in the simulation. We enhanced the Simulation framework implementation used by Baochun Li and Karen Wang[7].

In this implementation, the locations of the nodes are computed at the start of each simulation time unit using the mobility model specified as an input parameter. The nodes within the transmission range of each node are identified. Then, the simulation runs the service provisioning algorithm for all the nodes in the ad-hoc network. The simulation parameters are configurable through a configuration file. The implementation was done in C++.

The Simulation Framework had the implementation of Reference Point Group Mobility Model and Reference Velocity Group Mobility Model for testing the performance of the algorithm. We implemented our proposed algorithm and plugged it into the simulation framework. We also implemented the Random Direction Mobility Model along with its variations and used it for evaluating our algorithm within the framework.

4.3 Simulation Results and Analysis

4.3.1 Evaluation of Algorithm against different network densities

We varied the network density by varying the number of nodes in a fixed simulation area. We collected the statistics using the following values for the simulation parameters. We varied the number of nodes from 75 to 250 nodes with the simulation area measuring 750x750 distance units^2. All nodes followed the Reference Point Group Mobility model, and each node had a transmission range of 60 distance units. We chose the RPGM model for the study of the algorithm's behavior in a variable density network since the RPGM model had more frequent occurence of network partitioning than any other model. Wang and Li[6, 7] have used a nodal transmission range of 60 time units and a simulation area measuring 750x750 distance units^2 in their simulations. We use the same values for making a fair comparison. The simulations were run for 1000 time units under the assumption that reception intervals are equal to simulation time units.

Service Coverage

The ideal outcome of running the algorithm at all nodes in the network is that all nodes are reachable from the service instances running in the network. Fig.4.1 shows coverage of nodes in the network based on reachability from the service provider. This is to verify that the service instances running in the network provide adequate coverage. Fig.4.1 shows that the performance of the algorithm stabilizes after an initial delay. This is because of the propagation delay in the service instance information reaching all the nodes in the network. The initial delay is more pronounced as the number of nodes in the network increases. This is because we started with five initial service instances irrespective of the number of nodes in the network. As time t increases, the performance of the algorithm stabilizes.

The algorithm is based on each node's perspective of the network, i.e., each node selects one of the service instances as its service provider based on the information received from its neighbors. The Graph in Fig.4.2 shows the number of nodes that have identified a service instance, which is also available at that particular instant. We've verified the existence and the reachability of service instances in our simulation to determine the accuracy of the algorithm. Cross verification was warranted due to

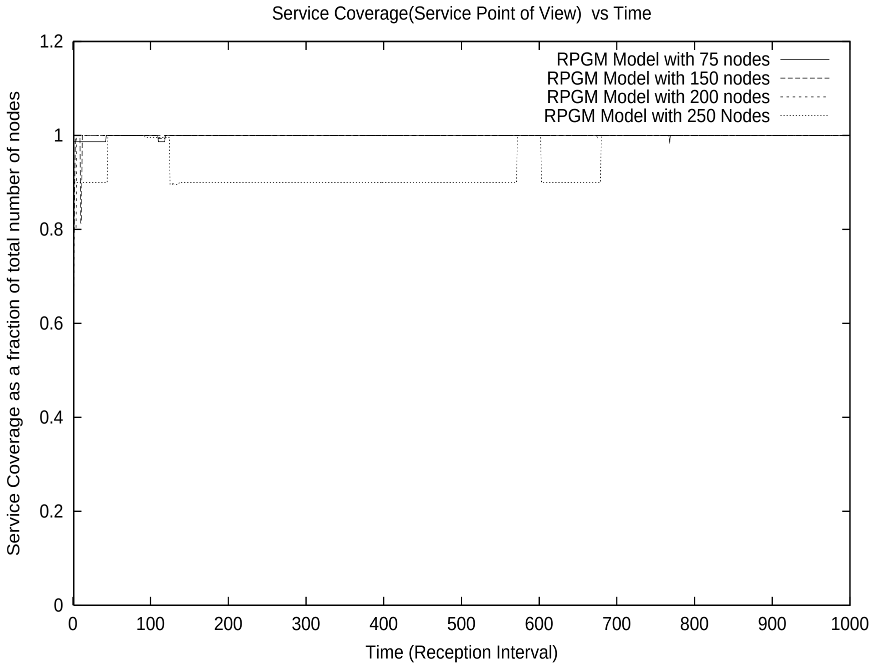


Figure 4.1: Graph showing Service Coverage (in terms of reachable nodes) under different network densities

the fact that there might exist stale information in the network due to propagation delay and network partitioning. We can see from the graph (Fig.4.2) that the algorithm's output closely follows the actual service coverage shown in graph (Fig.4.1).

Service Cost

The fraction of nodes in the network that had to host a service instance decreases as the density of the network increases especially when the number of nodes varies from 50 to 250(Fig.4.3). This is due to improved connectivity in the network resulting in a fewer disjoint groups in the network, hence the need for fewer service instances. The mean service cost of 10% obtained in our experimental runs is comparable to the results obtained for displacement based clustering algorithm as given in Li[6].

Frequency of Change in Service Provider Accessed by a Node

We measured how often a node changes its service provider. The frequency of change in service provider is given by the inverse of k where k is the number of time instants after which the node changes its service provider. So, as k increases, the frequency decreases. If a node changes its service

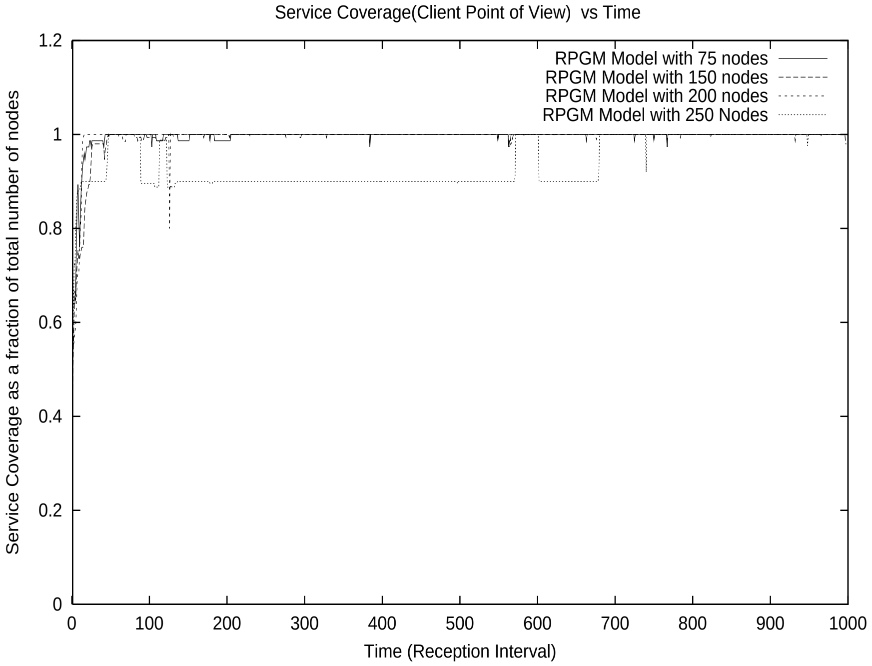


Figure 4.2: Graph showing Service Coverage (in terms of nodes that have identified accessible service instances) vs Time under different network densities

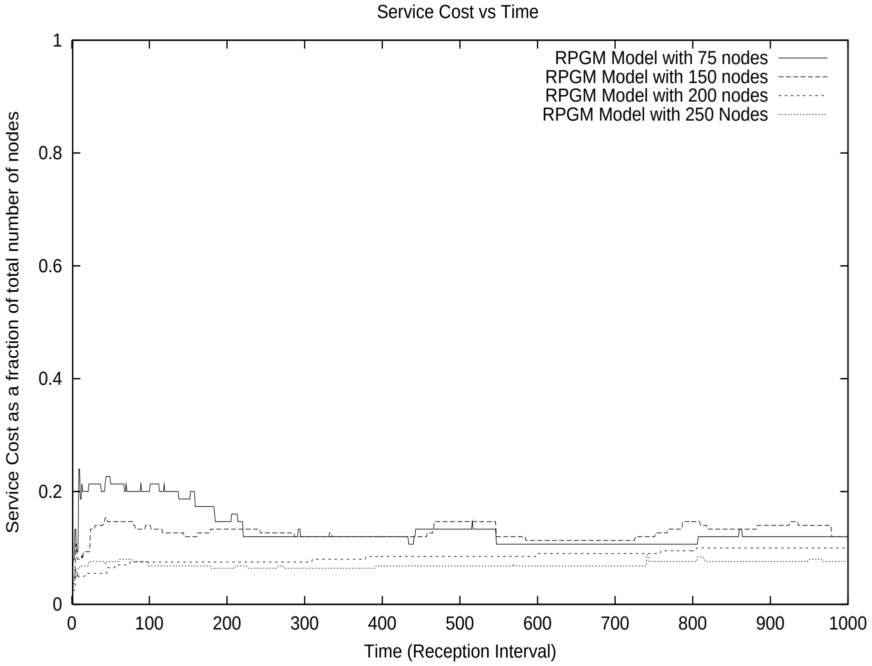


Figure 4.3: Graph showing Service Cost (as a fraction of total number of nodes) vs Time under different network densities

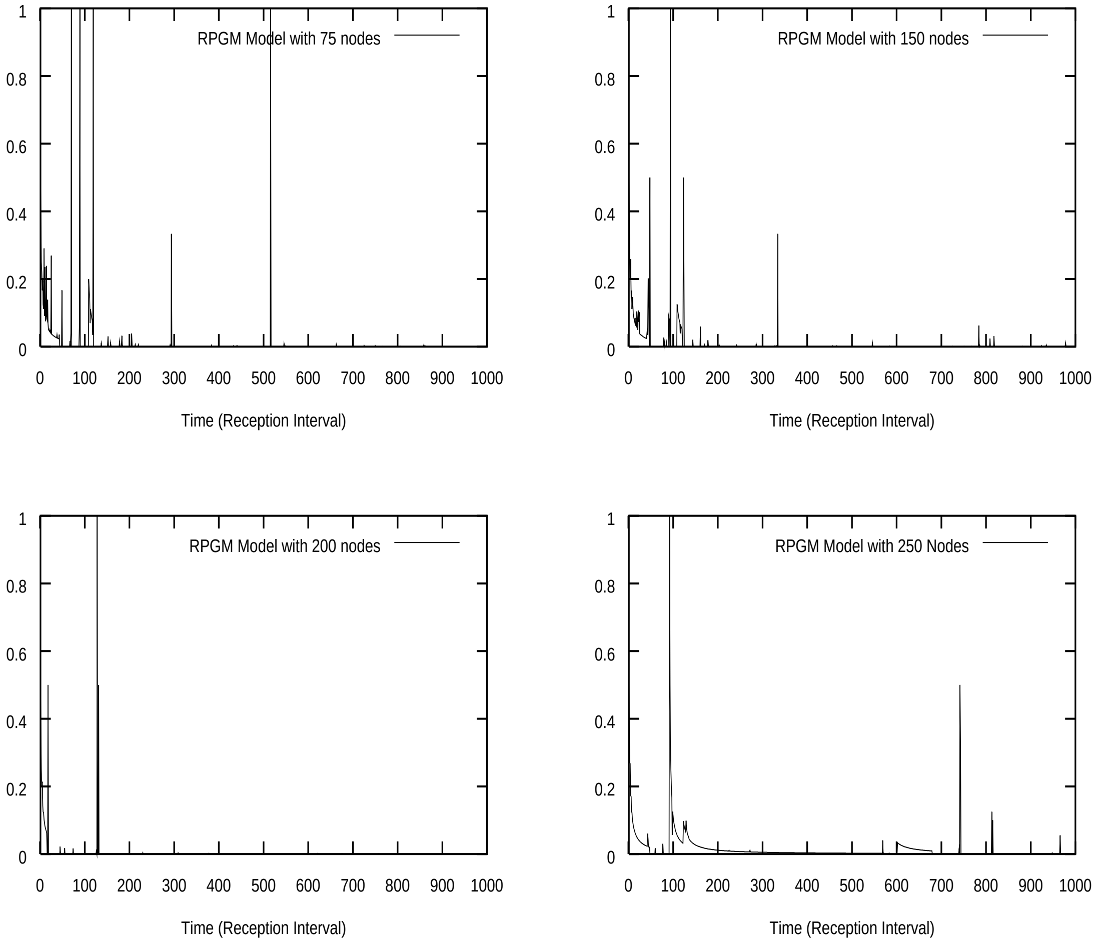


Figure 4.4: Graphs showing Frequency of Change in the service provider accessed by a node vs Time under different network densities

provider in the very next instant, the frequency is 1. Figure 4.4 shows the frequency of change in the service provider accessed by a node over a period of time. The values shown are the averages of the frequency of change in the accessed service provider of all the nodes that have undergone change in its accessed service provider at that instant of time.

It can be inferred that the algorithm is stable from the fact that as time increases, the frequency of change tends to 0. This can be seen from the curves falling down to zero for different network densities.

4.3.1.1 Experimental Summary

In short, the following observation were made based on the simulation results obtained by running the algorithm against varying network densities.

Service Coverage stabilized to a maximum value equal to the total number of nodes in the ad-hoc network after an initial delay equivalent to the propagation delay in service provider information across the network.

Service cost is inversely proportional to the network density. Service cost incurred by the non-clustering algorithm in the environment emulated by the RPGM model is the same as that incurred by the displacement based clustering algorithm which is 10% of the total number of nodes.

Frequency of change in the service provider accessed by a node stabilizes after an initial delay equivalent to the propagation delay in service provider information across the network. This holds good only for networks with considerable density. When the density of the network is low, the frequency of change is higher due to sparse connectivity of the network.

4.3.2 Evaluation of Algorithm against different Mobility Models

4.3.2.1 Experimental Configuration for all models

The simulation scenario involved running 150 nodes in a simulation area of 750x750 distance units^2 each with a transmission range of 60 distance units. The simulation was run under the assumption that each time instant is equal to one reception interval.

The Modified Random Direction Mobility Model was run in 8 different configurations. We used 4 basic configurations at two different range of speeds. Speed was normally distributed in the range of

0 to 20 distance/time units and 4 to 35 distance/time units in order to simulate slow and fast moving nodes, respectively. The four basic configurations were a combination of fixed pause-time and variable pause-time with bounded and boundless simulation area models. Fixed pause-time was set at 10 time units. Variable pause-time was varied in the range of [2,10] time units. All configurations had travel time varying between 2 and 10 time units, speed varying between 4 and 35 distance unit/time unit and direction Θ varying between 0 and 2π. Each node varied its travel-time, pause-time and velocity at the end of each cycle of travel time + pause-time. We chose these values for our simulation to simulate a significantly mobile ad-hoc network rather than a static network or a highly volatile network.

The Reference Point Group Mobility Model was run with 5 initial service instances and 2 concentrations each with 2 and 3 mobility groups. All the service instances were placed strategically at each concentration of mobile groups so that initially all mobile nodes can access a service instance.

The Hybrid Reference Velocity Group Mobility Model was run with 3 initial service instances in 3 configurations: 25% random, 50% random and 75% random. Degree of randomness is a measure of number of nodes in the network that follow the random walk model instead of the Hybrid RVGM model. The random walk was emulated by varying the node's speed between 0 and 10 distance/time units and changing direction uniformly between 0 and 2π. At the end of a mobility epoch, the nodes randomly select a new speed and direction. At the end of group mobility epoch, the group velocity changes. The mobility epoch used in the simulation was 30 time units, and the group mobility epoch was 90 time units.

Values for service coverage and service cost were obtained from the simulations for the above defined set of configurations.

4.3.2.2 Simulation Results

Based on the similarity of the results obtained in this simulation, we computed average of the results obtained for the random mobility models and chose 50% random Hybrid RVGM model for plotting the graphs(4.5, 4.6and 4.7).

Service Coverage:

From a service provisioning point of view, we measured service coverage in terms of number of nodes covered by at least one service instance in the adhoc network. Figure 4.5shows that the mean service coverage for the Reference Point Group Mobility Model is at maximum and equal to 1. This can be explained by the fact that the neighborhood of any node in the network does not change drastically during the course of time. As expected, the mean service coverage in the case of random mobility models ranges from 0.994 to 0.999, which is higher compared to the mean service coverage of the Hybrid RVGM model. This is because a higher degree of randomness gives rise to less occurrence of network partitioning. As a result, a node is mostly reachable by some service instance or the other in the case of the Modified Random Direction Mobility Model. In the case of the Hybrid RVGM model, the mean service coverage varies between 0.97 and 1.0.

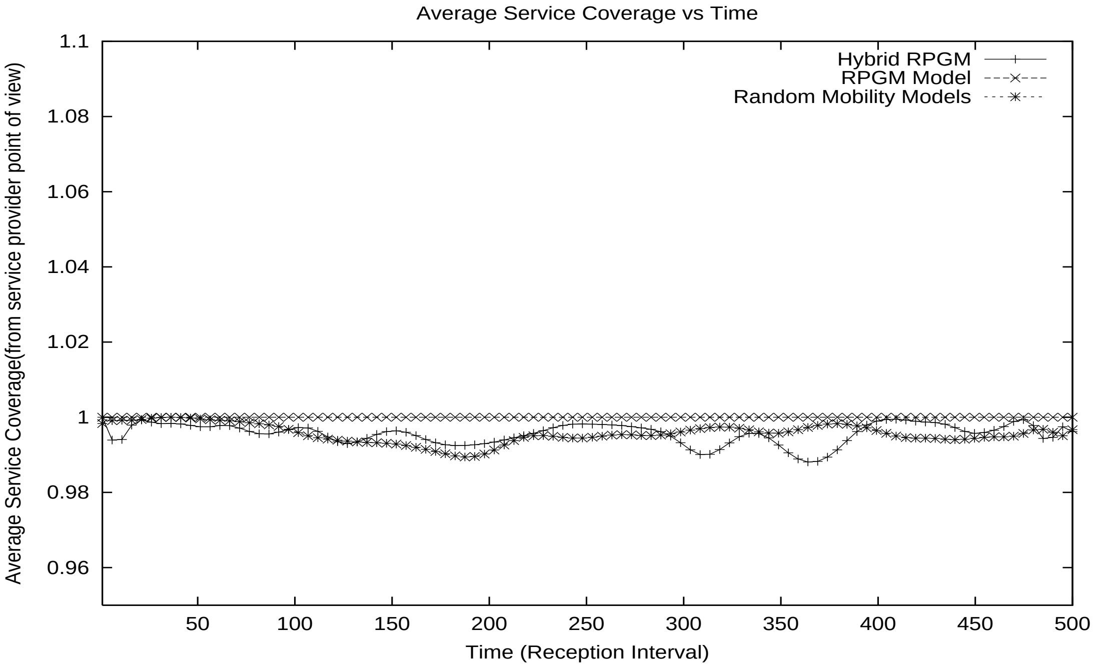


Figure 4.5: Graph showing Service Coverage (in terms of reachable nodes from service instances) vs Time for different mobility models

From the client's point of view, we measured the service coverage in terms of the number of nodes in the ad-hoc network that has identified a valid and an accessible service instance as its service provider. In this case, the mean service coverage for the Hybrid RVGM model, Modified Random Direction Mobility Model and RPGM model were 0.967, 0.978 and 0.995, respectively. All most all the nodes identify a valid service instance in the RPGM model as the algorithm quickly stabilizes due to stability in the neighborhood of the nodes.

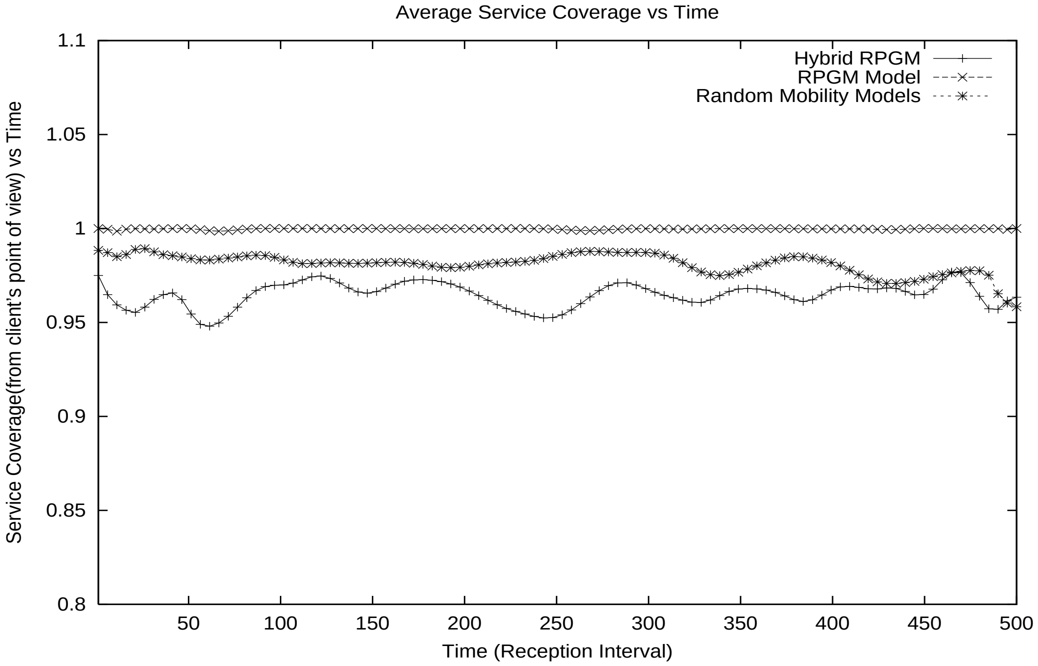


Figure 4.6: Graph showing Service Coverage (in terms of number of nodes that have accurately identified accessible service instances) vs Time for different mobility models

In the case of the Modified random direction mobility model and the Hybrid RVGM model, the mean service coverage is slightly lesser than the maximum service coverage possible(given by service coverage from the service point of view) in the network at that instant. This is because the propagation delay in the service instance information exceeds the rate of mobility of some of the nodes in the network and the neighborhood of a node in the hybrid RVGM model is more dynamic than that of a node in the RPGM model(Refer Fig.4.6). Moreover, propagation delay is not taken into consideration in Wang and Li methods for notification of service information among group members.

Service Cost

Service cost was estimated as the number of service instances running in the network to provide the above coverage. Figure 4.7 shows the service cost incurred in terms of number of service instances per disjoint group over time. From Figure 4.7, we can observe that the number of service instances required by the RPGM model is more than that incurred by any other model. This can be explained by the fact that the network is composed of a few large groups. There is frequent intermittent loss of connectivity between nodes with their immediate neighbors. As a result, the information propagation does not reach all nodes in the group at all times due to the effect of pruning information from ex-neighboring nodes. Since the proposed algorithm is based on the assumption that no node has knowledge of nodes beyond its transmission range except for the information gathered through information exchange subject to propagation delay, the number of service instances could not be optimized in terms of the number of disjoint set of connected components in the Graph G formed by the network nodes. This is a drawback compared to the performance of the algorithm proposed in [7]. The proposed algorithm performs well in the environments simulated by mobility models with some degree of randomness, as in the case of the Hybrid RVGM model and the Modified Random Direction Mobility Model where the mean service cost is 2 and 3, respectively. The service cost is compensated by the fact that a minimum of 95% service coverage is assured in all environments compared to the service coverage of velocity based clustering algorithm which falls to 70% to 80% in the case of random walk models[7].

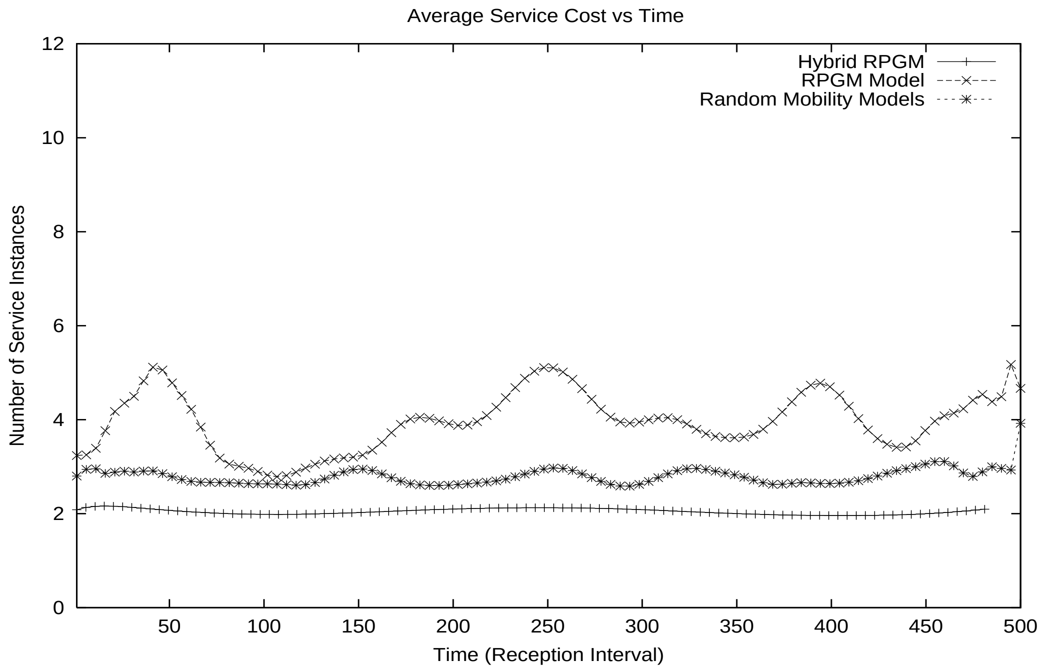


Figure 4.7: Graph showing Service Cost (as a fraction of number of disjoint sets of connected nodes) vs Time for different mobility models

Frequency of Migration of nodes from one service instance to another

A node in the ad-hoc network changes its service provider in order to have continuous access to the service. On the average, a node changes its service provider every 15 time units for the random mobility model, 5 time units for the Hybrid RVGM and once in 97 time units in the case of the RPGM model (refer Table 4.1). The frequency of change is more often in the case of the Hybrid RVGM because there is a high level of movement in the neighborhood of each node when the degree of randomness in the network is 50% and 75% whereas the frequency is comparatively lower when the network is 100% random (in case of random direction mobility model). Given the fact that the proposed algorithm incurs higher service cost in the RPGM model, the frequency of change in the service provider is low as expected.

Model	Modified Random Direction Mobility Model	RVGM	RPGM
Frequency of Migration	0.06643828	0.213838868	0.010390563

Table 4.1: Average Frequency of Migration of nodes from one service instance to another per instant of time

4.3.2.3 Experimental Summary

In short, the following observations were made based on the performance of the algorithm in the environment simulated by the three classes of mobility models.

Service coverage is high when the nodes in the network exhibit correlated movement with its neighboring nodes. Since this algorithm does not attempt to cluster the nodes in the network, the maximum service coverage comes with significant service cost. Since the network is more orderly than chaotic compared to other models, a node migrates from one service instance to another at an average rate of once in 97 time units.

The algorithm performs better when the nodes in the network move independently. In the environment where mobility is simulated by the Modified Random direction mobility model, the nodal movement is independent of each other. The number of groups in such a network does not change significantly, but group membership is highly dynamic. As a result, close to maximum service coverage is provided with limited service cost. In this environment, network partitioning and merging is a rare occurrence. As a result, a node is able to access a service provider for a significant period of time (i.e., 15 time units when the node with a transmission range of 60 distance units moves at the rate of 10 distance units/time unit in its chosen direction).

In the case of mobility simulated by the Hybrid Reference Velocity Group Mobility Model with varying degree of randomness, the algorithm performs at its best incurring a service cost of 1.8867 service instances per group to provide continuous uninterrupted service coverage but at the cost of frequent migration from one service provider to another. The frequent migration is due to the fact that the neighborhood of a node is highly dynamic.

4.3.3 Comparison of the Proposed Algorithm with the Velocity Based Clustering Algorithm

The algorithm proposed by Wang and Li[7] is based on clustering the nodes using velocity correlation. The algorithm was designed based on the Hybrid RVGM model. We compared the performance of the velocity based clustering algorithm and our proposed non-clustering algorithm in a mobility environment simulated by the Reference Velocity Group Mobility Model with varying degrees of randomness.

4.3.3.1 Experimental Configuration

We estimated the performance of both the algorithms in terms of service coverage and service cost in a network of 130 mobile nodes moving in a boundless simulation area of 750 * 750 distance units^2. All simulations were run for 2000 time units. We ran the simulation in 3 different configurations of the Hybrid RVGM model with 25%, 50% and 75% random mobility. The random walk was emulated by varying the node's speed between 0 and 10 distance/time units and changing direction uniformly between 0 and 2π. The mobility epoch used in the simulation was 30 time units, and the group mobility epoch was 90 time units. The simulation results obtained were consolidated to plot the graphs (refer Fig.4.8 and Fig.4.9).

4.3.3.2 Simulation Results

Service Coverage:

The service coverage provided by the proposed algorithm is at a maximum value equal to 1 irrespective of the degree of randomness in the movement of the nodes. The velocity clustering algorithm provides better service coverage when the nodal movement is highly random compared to complete correlated movement. This is due to the fact that there's lesser number of network partitioning in a chaotic environement. Even then, the maximum service coverage obtained in the 75% random Hybrid RVGM model is 70%. This is because detection of long term groups returns inaccurate results when the network is highly dynamic. The mean service coverage provided by the Velocity based clustering algorithm is 60% compared to 99.95% service coverage provided by our proposed algorithm (Fig.4.8). This shows that our algorithm addresses the problem better in a mobility environment where there's more randomness than group mobility. The result obtained can be explained by the fact that the clustering algorithm aims at minimizing service instances in a group more than providing continuous uninterrupted service coverage to all the nodes in the network.

Service Cost

The graph showing change in Service Cost for the two algorithms over a period of time is given in Figure 4.9. The non-clustering algorithm incurs a mean service cost of 1.52, 1.89 and 2.25 service instances per disjoint group for 75%, 50% and 25% randomness, in the Hybrid RVGM model respectively. The velocity based clustering algorithm incurs a mean service cost of 0.3, 0.35 and 0.42 service instances per disjoint group for 25%, 50% and 75% randomness in the Hybrid RVGM model respectively.

The service cost incurred by the velocity based clustering algorithm is less around 0.36 times the number of disjoint groups in the network whereas our algorithm incurs a higher cost of 1.8867 service instances per group. In an attempt to minimize the service cost, the clustering algorithm regards the digression of some nodes, from group movement, as temporary and keeps the number of service instances equivalent to the number of groups identified using velocity correlation. The number of groups identified by the velocity based clustering algorithm is less than the number of disjoint set of

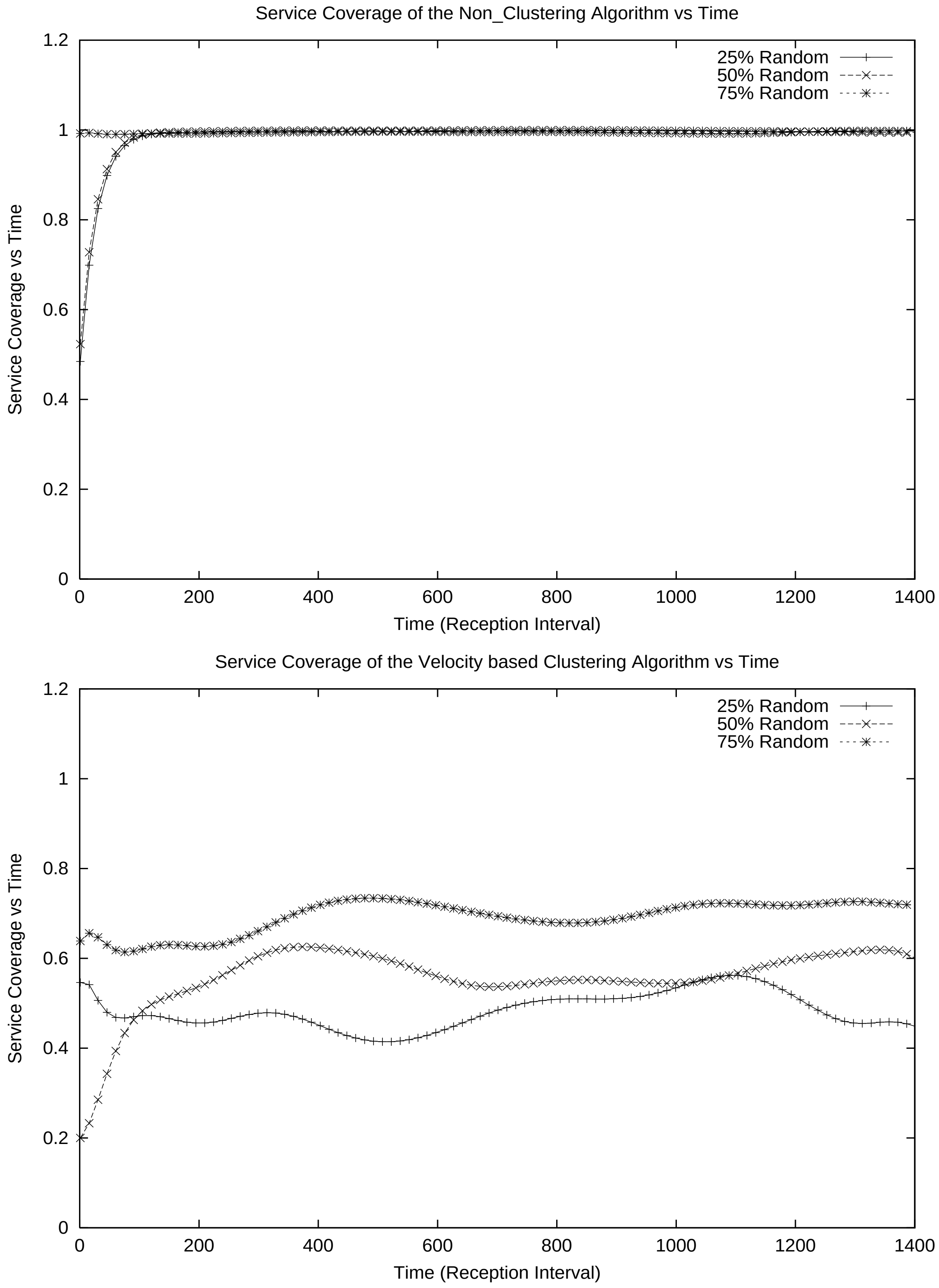


Figure 4.8: Graphs showing Service Coverage provided by the proposed algorithm and the velocity based clustering algorithm

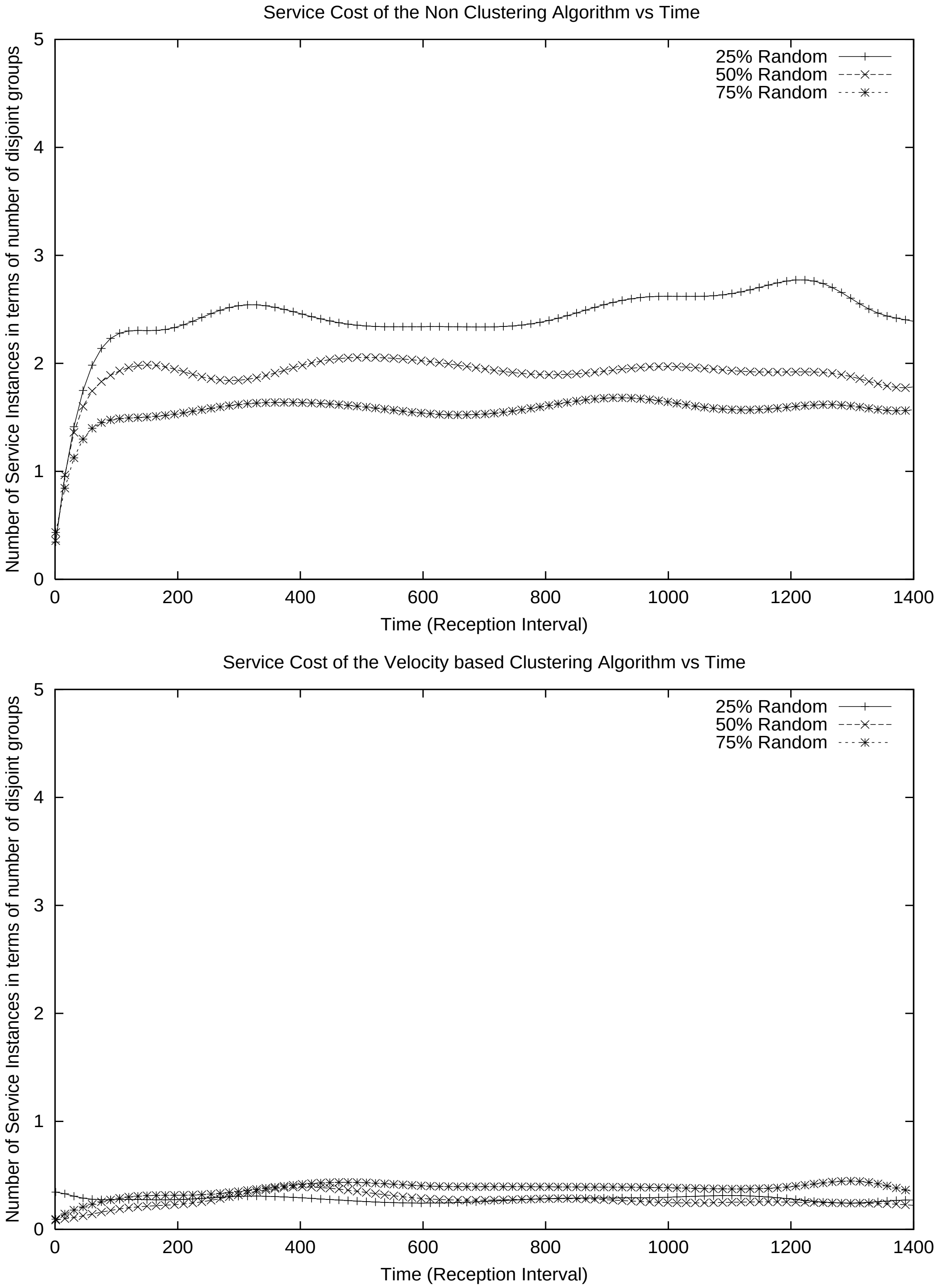


Figure 4.9: Graphs showing Service Cost incurred by the proposed algorithm and the velocity based clustering algorithm

connected components in the network at that instant of time. From the results given in Wang and Li[7], service coverage of 80% to 90% is achieved at a mean service cost of 1.35 service instances.

4.3.3.3 Experimental Summary

The proposed algorithm provides maximum service coverage of 99.95% at the cost of 1.8867 service instances per disjoint set of connected nodes. The velocity based clustering algorithm provides efficient service coverage with 60% service coverage at the cost of 0.36 service instances per group.

4.3.4 Conclusion

In this chapter, we evaluated the performance of the algorithm against varying network densities of the network and against different mobility models. We found that the proposed algorithm performs better in a mobile environment where there's more random movement compared to environments where there's only correlated movement. We compared the performance of our proposed algorithm against the velocity based clustering algorithm proposed by Wang[7]. The non-clustering algorithm provides maximum continuous service coverage compared to the clustering algorithm[7] but at a higher service cost. Depending on how critical the provisioning of service is compared to the optimization of the cost, a network-wide service deployer can choose between the non-clustering algorithm and the clustering algorithm, respectively. In the next chapter, we will explore ways to improve the algorithm and conclude by quantifying the results obtained as a result of this research work.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In the scenario defined in Section 1.5, guaranteed connection to a database server was of a higher priority compared to minimizing the number of database servers though both are significant objectives of the expected solution. Hence, we defined the goal of our research work as providing continuous and guaranteed service coverage while minimizing the number of service instances in the adaptive services-aware ad-hoc networks. We then proposed a strategy which does not attempt to cluster the nodes in the ad-hoc network but provides a generalized solution to the problem independent of any mobility model. We proposed the usage of simple message-passing techniques in conjunction with a distributed localized algorithm for providing the necessary service coverage in the ad-hoc networks.

We then derived the complexity of the algorithm in terms of communication, messaging and information processing. Communication complexity was proved to be O(n^2) where n is the average number of nodes in a connected component in Graph G formed by the nodes in the ad-hoc network. The messaging complexity was proved to be O($n\log N$) in the worst case where n is the average number of nodes in a connected component and N is the total number of nodes in the ad-hoc network. The computational complexity of the algorithm was proved to be O($k\log k$) or O(s_g) whichever is greater, where k is the number of neighbors of a node and s_g the number of service instances in the group in which the node resides.

Since the mobility pattern of the elements in the battlefield is uncertain, we evaluated the proposed solution against different classes of mobility models. We found that the algorithm provides maximum service coverage but at a service cost on the order of the number of connected components in the graph formed by the nodes in the network. We found that the algorithm's performance converged after an initial delay equivalent to propagation delay in the order of the depth of the spanning tree rooted at the service instance hosting node. The nodes did not change the service instance accessed for a significant period of time. This also corresponded to less change in the node set hosting service instances. This proved that the algorithm performs optimal placement of service instances in the ad-hoc network.

The algorithm did not perform well compared to the velocity based clustering algorithm in terms of the service cost incurred when the nodes followed the RPGM model with frequent intermittent loss of connectivity between nodes. This also prevented the usage of link stability information as a part of the choice of service provider for improving the performance of the algorithm. We presented the simulation results obtained to evaluate the performance of the algorithm against the mobility models and compared its performance with the velocity based clustering algorithm[7] in chapter 4.

5.2 Future Work

The information propagation has to be optimized further to improve the sensitivity of the algorithm. Not all kinds of looping of information have been removed from the information flow. This does not affect the performance of the algorithm when the nodal movement is more random than group oriented. Its adverse affect is seen significantly when nodes move slowly and in groups. When the nodes move slowly, the information tends to stay longer in the network and hence the effect of indirect looping is more prominently seen. That's the reason why we've significant difference in the service coverage measured from the service point of view (0.985) and the service coverage measured from the client's perspective in the RVGM model(0.967) (refer Section 4.3.2.2).

Consider Figure 5.1. Communication from nodes 6 and 5 to node 1 occurs for passing information about the path to service instance at node 0 through nodes 5 and 6, respectively. Actually the alternate path is not another possibility for node 1 to access node 0 as the alternate paths are extensions of the path from node 0 to node 1. This information is inaccurate in influencing the accessibility decision of service instance at node 1. If we record the path sequence as the information gets propagated, we can

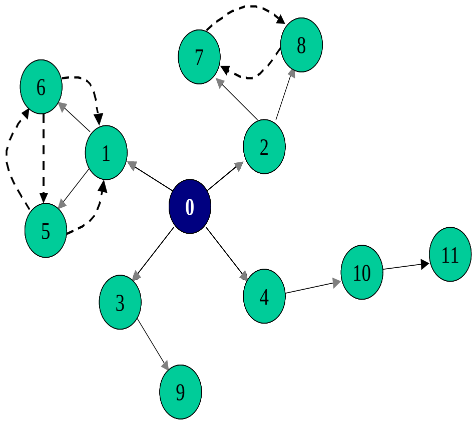


Figure 5.1: Redundant Information Propagation from nodes 5 and 6 to node 1

reduce this looping and make the algorithm more responsive.

There are many flooding optimization techniques available in literature. Reduction in looping will greatly enhance the sensitivity of the algorithm. The communication complexity of the algorithm will drastically reduce to $O(N_g)$ where N_g is the total number of nodes in a group, and we estimate the service cost also to reduce significantly as a result of this change.

A hybrid of the clustering algorithm[7] and our non-clustering algorithm will better address the problem of providing continuous service coverage at a reduced service cost in an environment where there's group mobility with a significant degree of randomness. Our algorithm emphasizes providing continuous service coverage to all nodes in the ad-hoc network, whereas the velocity based clustering algorithm aims at minimizing the service cost.

Bibliography

[1] Gnutella. http://gnutella.wego.com. Accessed January 8, 2003.

[2] Napster. http://www.napster.com. Accessed January 8, 2003.

[3] Project JXTA. http://wwws.sun.com/software/jxta/jxta-doc.html. Accessed January 8, 2003.

[4] J. M. Kahn, R. H. Katz and K. S. J. Pister, "Mobile Networking for Smart Dust," ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 99), Seattle, WA, August 17-19, 1999

[5] Wireless Integrated Network Sensors project website. http://www.janet.ucla.edu/WINS/

[6] Baochun Li, "QoS-aware Adaptive Services in Mobile Ad-hoc Networks," in *Proceedings of the Ninth IEEE International Workshop on Quality of Service (IWQoS 01)*, also *Lecture Notes in Computer Science*, ACM Springer-Verlag, Vol. 2092, pp. 251-268, Karlsruhe, Germany, June 6-8, 2001.

[7] Karen Wang, Baochun Li, "Efficient and Guaranteed Service Coverage in Partitionable Mobile Ad-hoc Networks," in *Proceedings of IEEE INFOCOM 2002*, New York City, New York, June 23-27, 2002.

[8] S. Jiang, D. He, and J. Rao, "A Prediction-based Link Availability Estimation for Mobile Ad Hoc Networks," in *Proceedings of IEEE INFOCOM*, pp.1745-52, April 2001. http://citeseer.nj.nec.com/498937.htm

[9] S. Basagni, “Distributed and Mobility-Adaptive Clustering for Multimedia Support in Multi-Hop Wireless Networks,” in *Proceedings of Vehicular Technology Conference*, VTC 1999-Fall, vol. 2, pp. 889-893. http://citeseer.nj.nec.com/article/basagni99distributed.html

[10] J. Gao, L. J. Guibas, J. Hershberger, L. Zhang, and A. Zhu, “Discrete mobile centers,” in *Proceedings of 17th ACM Symposium on Computational Geometry*, June 2001.

[11] B. Liang and Z. Haas, “Predictive distance-based mobility management for PCS networks,” in *Proceedings of the Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, March 1999.

[12] M. Gerla and J. Tsai, “Multicluster, mobile, multimedia radio network," *ACM-Baltzer Journal of Wireless Networks*, Vol. 1, No. 3, pp. 255-265, 1995.

[13] W. Su, S.J. Lee, and M. Gerla, “Mobility Prediction in Wireless Networks," in *Proceedings of the IEEE Military Communications Conference (MILCOM)*, Los Angeles, CA, October 2000.

[14] Jonathan Chan and Aruna Seneviratne, “A Practical User Mobility Prediction Algorithm for Supporting Adaptive QoS in Wireless Networks,” in *IEEE International Conference on Networks 99*, January 1999.

[15] Tracy Camp, Jeff Boleng and Vanessa Davies, “A Survey of Mobility Models for Ad hoc network Research,” *Wireless Communications & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, Vol.2, No.5, pp.483-502, 2002.

[16] C.K. Toh, “Associativity-based routing for ad hoc networks," *Wireless Personal Communications*, Mar 1997, pp.103-139.

[17] R. Dube, C. Raia, K-Y Wang and S. Tripathi, “Signal stability based adaptive routing (SSA) for ad hoc networks,” *IEEE Personal Communications*, Feb. 1997, pp. 36-45.

[18] A. Bruce McDonald and T.F. Znabi, “A path availability model for wireless ad hoc networks,” in *Proceedings of IEEE Wireless Communications and Networking Conference,* New Orleans, USA, (Sept. 1999), pp. 35-40.

[19] R. Ramanathan and M. Steenstrup, “Hierarchically-organized, multihop mobile wireless networks for quality-of-service support,” *Mobile Networks & Applications*, vol. 3, pp. 101-119, June 1998.

[20] S. Basagni, “Distributed clustering for ad hoc networks,” in *Proceedings of the 1999 International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN’99)*, pp. 310-315, (Perth/Fremantle, Australia), IEEE Computer Society, June 1999.

[21] Z. Haas and M. Pearlman, “ The zone routing protocol (ZRP) for ad hoc networks,” Internet Draft(work in progress), July 2002.

[22] C. R. Lin and M. Gerla, “Adaptive clustering for mobile wireless networks,” *IEEE Journal of Selected Areas in Communications*, Vol. 15, pp. 1265-1275, September 1997.

[23] Mainak Chatterjee, Sajal K. Das, and Damla Turgut. “WCA: A weighted clustering algorithm for mobile ad hoc networks,” *Journal of Cluster Computing (Special Issue on Mobile Ad hoc Networks)*, vol. 5, pp. 193-204, April 2002.

[24] L. E. Miller, July 2000. “Flooding,” compilation of MANET mail list. http://w3.antd.nist.gov/wctg/manet/docs/flooding.pdf. (25.November.2002).

[25] Christopher Ho, Katia Obraczka, Gene Tsudik, and Kumar Viswanath, “Flooding for reliable multicast in multi-hop ad hoc networks,” in *Proceedings of the 3rd International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, pp. 64-71, Seattle, WA, 1999.

[26] Kumar Viswanath and Katia Obraczka, “An Adaptive, Integrated Approach to Reliable Group Communications in Multi-Hop Ad hoc Networks,” in *IEEE Symposium on Computers and Communications (ISCC 2002).*

[27] B. Williams and T. Camp, “Comparison of broadcasting techniques for mobile ad hoc networks,” in *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, pages 194-205, 2002.

[28] H. Lim and C. Kim, “Multicast tree construction and flooding in wireless ad hoc networks,” in *Proceedings of the ACM International Workshop on Modeling*, Analysis and Simulation of Wireless and Mobile Systems (MSWIM), 2000.

[29] Goutham Karumanchi, Srinivasan Muralidharan, and Ravi Prakash, “Information dissemination in partitionable mobile ad hoc networks,” in *Symposium on Reliable Distributed Systems*, pp. 4-13, 1999.

[30] Christos Efstratiou, Keith Cheverst, Nigel Davies, Adrian Friday, “An Architecture for the Effective Support of Adaptive Context-Aware Applications,” in *Mobile Data Management* 2001, pp. 15-26.

[31] Joanna Kulik, Wendi Rabiner, Hari Balakrishnan, “Adaptive Protocols for Information Dissemination in Wireless Sensor Networks,” in *Proceedings of 5th ACM/IEEE Mobicom Conference,* Seattle, WA, August 1999.

[32] M. M. Zonoozi and P. Dassanayake, “User mobility modeling and characterization of mobility patterns,” *IEEE Journal on Selected Areas in Communications,* vol.15(7), pp. 1239-1252, September 1997.

[33] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu and Jorjeta Jetcheva, “A performance comparison of multi-hop wireless ad hoc network routing protocols,” *the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking: ACM (MOBICOM’98),* Dallas TX, October 1998.

[34] Elizabeth M. Royer, P. Michael Melliar-Smith, and Louise E. Moser, “An Analysis of the Optimum Node Density for Ad hoc Mobile Networks,” in *Proceedings of the IEEE International Conference on Communications,* Helsinki, Finland, June 2001.

[35] V. Tolety, “Load reduction in ad hoc networks using mobile servers,” Master’s thesis, Colorado School of Mines,1999.

[36] A. Bruce McDonald and T.F. Znabi, “A mobility-based framework for adaptive clustering in wireless ad hoc networks,” *IEEE Journal in Selected Areas of Communication*, vol.17(8), pp.1466-1487, August 1999.

[37] K. Wang, B. Li, “Group Mobility and Partition Prediction in Wireless Ad-hoc Networks ,” in *Proceedings of IEEE International Conference on Communications (ICC 2002),* New York City, New York, April 28 - May 2, 2002.

[38] L. E. Miller, July 2000. “Overhead Analysis," compilation of MANET mail list. http://w3.antd.nist.gov/wctg/manet/docs/overhead.pdf. (25.November.2002).

[39] Andy Harter, Andy Hopper, Pete Steggles, Andy Ward, and Paul Webster, “The anatomy of a context-aware application,” in *Proceedings of the fifth annual ACM/IEEE international conference on Mobile computing and networking,* pp. 59-68. ACM Press, 1999.

[40] Noble, B., Satyanarayanan, M., Nguyen, G., and Katz, R., “Trace-Based Mobile Network Emulation,” *Computer Communication Review,* Vol. 27, No.4, October 1997, pp. 51-61.

[41] Magnus Frodigh, Per Johansson and Peter Larsson, “Wireless ad hoc networking- the art of networking without a network,” *Ericsson Review,* No.4, 2000. pp.248-263.

[42] Elizabeth M. Royer and Chai-Keong Toh, “A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks,” *IEEE Personal Communications,* April 1999. pp.46-55.

[43] M. W. Subbarao, 15 September 1999, “Ad Hoc Networking Critical Features and Performance Metrics”. http://w3.antd.nist.gov/wctg/manet/docs/critical_features.pdf. (25.November.2002).

[44] Kostas P. Hatzis, George P. Pentaris, Paul G. Spirakis, Vasilis T. Tampakas and Richard B. Tan, “Fundamental Control Algorithms in Mobile Networks,” in *Proceedings of 11th Annual ACM Symposium on Parallel Algorithms and Architectures*, pp. 251-260, 1999. http://citeseer.nj.nec.com/hatzis99fundamental.html.

[45] David Eppstein, “Finding the k shortest paths”, Proc. 35th Symp. Foundations of Computer Science, IEEE, pp.154-165, 1994.

[46] E.Q.V. Martins, “An Algorithm for Ranking Paths that May Contain Circles,” European Journal of Operational Research, 18 (1984), pp.123-130.

[47] E.Q.V. Martins, M.M.B. Pascoal, and J.L.E. Santos, “Deviation Algorithms for Ranking Shortest Paths,” International Journal of Foundations of Computer Science, Vol. 10, No. 3 (1999), pp247-261.

[48] V.M. Jimenez and A. Marzal, “Computing the K Shortest Paths: a New Algorithm and an Experimental Comparison,” Springer-Verlag, Lecture Notes in Computer Science Series, Vol. 1688, pp.15-29, 1999.

[49] J. W. Suurballe, “Disjoint Paths in a Network,” Networks, 4 (1974), pp.125-145.

[50] D. Torrieri, “Algorithms for Finding an Optimal Set of Short Disjoint paths in a Communication Networks,” IEEE Transactions on Communications, Vol. 11, Nov. 1992, pp.1698-1702.

[51] M.Sanchez, “Mobility Models,” http://www.disca.upv.es/misan/mobmodel.htm. (7.August.2002).

[52] Cheverst, K., N. Davies, K. Mitchell and A. Friday, “Experiences of Developing and Deploying a Context-Aware Tourist Guide: The GUIDE Project,” in *Proceedings of MOBICOM’2000*, Boston, ACM Press., 2000.