INSTITUTO TECNOLOGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY

CAMPUS MONTERREY

DIVISION DE ELECTRONICA, COMPUTACION, INFORMACION, Y COMUNICACIONES

PROGRAMA DE GRADUADOS

EVALUATING CONNECTIVITY AND QUALITY IN AD-HOC NETWORKS THROUGH CLUSTERING AND TRELLIS ALGORITHMS

THESIS

PRESENTED AS A PARTIAL FULFILLMENT OF THE
REQUERIMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN ELECTRONIC ENGINEERING
MAJOR IN TELECOMMUNICATIONS

MARTHA LUCIA TORRES LOZANO

MONTERREY, N.L. MAY 2003

Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Monterrey

División de Electrónica, Computación, Información, y Comunicaciones

Programa de Graduados

Evaluating Connectivity and Quality in Ad-Hoc Networks through Clustering and Trellis Algorithms

THESIS

Presented as a partial fulfillment of the requeriments for the degree of

Master of Science in Electronic Engineering

Major in Telecommunications

Martha Lucia Torres Lozano

Monterrey, N.L. May 2003

Acknowledgments

I thank Instituto Tecnológico y de Estudios Superiores de Monterrey for the facilities provided. I am grateful to my thesis advisor, César Vargas Rosales, Ph.D., because this thesis could not have been possible without his interest and collaboration. I mention specially to Jose Ramón Rodriguez Cruz, Ph,D., and Artemio Aguilar Coutiño, M.Sc., for their comments that have helped to enhance this work.

My friends, Catalina, Beatriz, Carolina, Ericka, Franklin, Juan D., Fernando, Perla, Rafael and all my friends and partners from Colombia, México and CET for their friendship and helping in this period of time.

MARTHA LUCIA TORRES LOZANO

Instituto Tecnológico y de Estudios Superiores de Monterrey
May 2003

To my parents, Federico and Julia
To my siblings, Diana and Lina
To my boyfriend, Eric
For their love, guidance, inspiration,
patience and support.

Evaluating Connectivity and Quality in Ad-Hoc Networks through Clustering and Trellis Algorithms

Martha Lucia Torres Lozano, M.Sc.
Instituto Tecnológico y de Estudios Superiores de Monterrey, 2003

Abstract

Recently, wireless networks have become increasingly popular in the computing industry. These networks provide mobile users with ubiquitous computing capability and information access regardless of the location. There are currently two variations of mobile wireless networks- infrastructured (e.g., cellular network) and "infrastructureless" networks, called Mobile Ad-Hoc Networks, where the entire network is mobile, and the individual terminals are allowed to move at will, relative to each other, then they are self-creating, self-organizing, and self-administering.

In order to improve the users service, there exist some measures as connectivity, quality, throughput, and others, to permit evaluate the performance in the network. In this work, we evaluate connectivity and quality using clustering algorithms, DDCA (Distributed Dynamic Clustering Algorithm), and Trellis Method to find k-paths to the different users. Therefore, we simulate an ad-Hoc network, define clusters and parameters as in the DDCA algorithm, and apply concepts about connectivity and quality, depending on parameters defined in the network simulation, such as amount of users and clusters, link availability probabilities, and others.

Contents

List of Figures iii

List of Tables v

Chapter 1 Introduction 1

1.1 Objective 2
1.2 Justification 2
1.3 Contribution 3
1.4 Organization 3

Chapter 2 Wireless Ad-Hoc Networks 5

2.1 Important Aspects about Ad-Hoc Networks 6
2.1.1 Traffic Profiles 6
2.1.2 Types of Ad-Hoc Networks 7
2.1.3 Security and Privacy 7
2.2 Characteristics of MANETs 7
2.3 Routing in Ad-Hoc Networks 8
2.4 Ad-hoc Mobile Routing Protocols 10
2.4.1 Table-Driven Approaches 11
2.4.2 Source-Initiated On-Demand Approaches 14
2.5 Quality of Service (QoS) 17

Chapter 3 Clustering in Ad-Hoc Networks 19

3.1 Clustering Algorithm 20
3.1.1 DMAC Algorithm 20
3.1.2 ABCP Algorithm 21
3.1.3 Application of the clustering algorithms to the sample scenario . . . 23
3.2 DDCA Algorithm 26
3.2.1 (α, t) Cluster Framework 27
3.2.2 (α, t) Cluster Characterization 28

3.2.3 Clustering Algorithm . . . 29
3.3 Trellis Algorithm for Ad-Hoc Networks . . . 33
3.3.1 Graph Modeling . . . 33
3.3.2 Link and Path Cost . . . 35
3.3.3 Algorithm Net to Trellis . . . 36
3.3.4 Example applying Trellis Algorithm . . . 39
3.4 Connectivity and Quality of Network . . . 47

Chapter 4 Model Description **49**
4.1 Network Generation . . . 49
4.2 Connectivity and Quality Application . . . 51

Chapter 5 Numerical Results **57**
5.1 Network Generation . . . 57
5.1.1 Cluster Generation . . . 57
5.1.2 Link Generation . . . 57
5.2 Proposed Scenarios . . . 59
5.2.1 Scenario 1 . . . 59
5.2.2 Scenario 2 . . . 62
5.2.3 Scenario 3 . . . 71
5.2.4 Scenario 4 . . . 77

Chapter 6 Conclusion and Further Research **85**
6.1 Conclusions . . . 85
6.2 Further Research . . . 87

Bibliography **89**

Vita **91**

List of Figures

2.1 Example of a simple Ad-Hoc Network with three participating nodes . . . 6
2.2 Block Diagram of a mobile node acting both as hosts and as router 6
2.3 An Ad-Hoc Network example 8
2.4 Schemes for routing in Ad-Hoc Networks 9
2.5 Categorization of Ad-Hoc Routing Protocols. 11
2.6 A CSGR path. 12
2.7 Routing Inefficiency in CGSR. 13
2.8 ZRP and Concept of Bordercasting. 16

3.1 Example of cluster of Ad-Hoc network 20
3.2 DCMA Flowchart 22
3.3 ABCP Flowchart 23
3.4 Initial Ad hoc Network 24
3.5 a) New Node, DMAC Algorithm. b)New Node, ABCP Algorithm. 25
3.6 a) Link Failure, DMAC Algorithm. b)Link Failure, ABCP Algorithm. . . . 25
3.7 a) New Link, DMAC Algorithm. b) New Link, ABCP Algorithm. 26
3.8 General flowchart for DDCA Algorithm 27
3.9 Notation to Definition 1. 29
3.10 Node Activation Flowchart 31
3.11 Link Activation Flowchart 32
3.12 Link Failure and Node Deactivation 34
3.13 A K-Trellis graph with L=5, H=4 35
3.14 Illustration of application of the Operation $P1$ and $P2$. 38
3.15 Trellis Algorithm Flowchart 39
3.16 Transformation from normal network to Trellis Graph 40
3.17 Initial Ad-Hoc Network. 41
3.18 Applying the Trellis Algorithm in Cluster A. 41
3.19 Trellis Algorithm applied to cluster B 42
3.20 Trellis Algorithm applied into cluster C 42

3.21 Trellis Algorithm applied taking clusters as a nodes 42
3.22 Trellis Algorithm applied in all the network 43
3.23 Paths with minor probabilities. 44
3.24 Link Failure Scenario. 45
3.25 New Node Scenario. 46
3.26 New link Scenario. 46
3.27 Connectivity example. 47
3.28 a) High Density of nodes, b) Low density of Nodes. 48

4.1 Illustration of nodes, distances and Pr_{ij} of (4.3) 51
4.2 Network Generation Algorithm 52
4.3 Types of Connectivity at the network generated 52
4.4 Types of Quality at the network generated 54

5.1 Organization area in different numbers of cluster 58
5.2 Network Connected with 4 Clusters 58
5.3 Initial Network 60
5.4 Situations Scenario 1 61
5.5 Performance into the Cluster(4 Clusters) 63
5.6 Performance into the Cluster (9 Clusters) 64
5.7 Performance into the Cluster (16 Clusters) 65
5.8 Total Cluster Performance for 100 nodes 66
5.9 Total Cluster Performance for 200 nodes 67
5.10 Total Cluster Performance for 400 nodes 67
5.11 $TotalC_{BL}$ and $TotalQ_{BL}$ for 100 nodes 68
5.12 $TotalC_{BL}$ and $TotalQ_{BL}$ for 200 nodes 69
5.13 $TotalC_{BL}$ and $TotalQ_{BL}$ for 400 nodes 70
5.14 Total Connectivity and Quality in the network 73
5.15 Connectivity and Quality per Cluster 73
5.16 Connectivity and Quality into the Cluster. 75
5.17 Total Connectivity and Quality respect Link Borders 76
5.18 Total Connectivity and Quality in the network 79
5.19 Connectivity and Quality per Cluster 79
5.20 Connectivity and Quality into the Cluster. 81
5.21 Total Connectivity and Quality with respect to Link Borders 83

6.1 Trellis and DDCA Algorithm Application 86

List of Tables

5.1 Connectivity and Quality to Scenario 1, percentage 60
5.2 Connectivity and Quality if a Node is added to the Network, percentage . . 61
5.3 Connectivity and Quality if a Link Fails, percentage 61
5.4 Connectivity and Quality if a Link is added, percentage 61
5.5 Total Cluster Performance, percentage 62
5.6 Parameters to Scenario 2 62
5.7 Parameters to Scenario 3 71
5.8 Total Connectivity and Quality, in percentage 71
5.9 Connectivity and Quality per Cluster, in percentage 72
5.10 Connectivity and Quality into the Cluster with respect to NC, in percentage 74
5.11 Performance into the Cluster respect the Realization Number, in percentage 74
5.12 Cluster Performance respect to the Link Border Probabilities, in percentage 76
5.13 Parameters to Scenario 4 77
5.14 Total Connectivity and Quality, in percentage 77
5.15 Connectivity and Quality per Cluster, in percentage 78
5.16 Connectivity and Quality into the Cluster with respect to NC, in percentage 80
5.17 Performance into the Cluster respect the Realization Number, in percentage 80
5.18 Cluster Performance respect to the Link Border Probabilities, in percentage 82

6.1 Acceptable Values to Connectivity and Quality 87

Chapter 1

Introduction

Since their emergence in 1970's, wireless networks have become increasingly popular in the computing industry. These networks provide mobile users with ubiquitous computing capability and information access regardless of the location. There are currently two variations of mobile wireless networks- infrastructured and "infrastructureless" networks.

The infrastructured networks (e.g., cellular network), have fixed, wired gateways and centralized administration for their operations. They have fixed base stations which are interconnected to other base stations. The transmission range of a base station constitutes a cell. All the mobile nodes lying within this cell connect to and communicate with the nearest bridge (base station), [1]. In contrast, infrastructureless networks, called Mobile Ad-Hoc Networks (MANET), do not have fixed routers. All nodes are capable of movement and can be connected dynamically in an arbitrary manner. The responsibilities for organizing and controlling the network are distributed among the terminals themselves, in other words, Ad Hoc wireless networks are self-creating, self-organizing, and self-administering. The entire network is mobile, and the individual terminals are allowed to move at will, relative to each other. The nodes for this class of network can be located in or on airplanes, ships, trucks, cars, perhaps even on people or very small devices like laptops.

Communication, between arbitrary endpoints in ad hoc networks, requires routing over multiple hops wireless-hop paths. The main difficulty arises because without a fixed infrastructure, these paths consist of wireless links whose endpoints are likely to be moving independently of one another. Consequently, node mobility causes the frequent failure and activation of links which leads to increased network congestion, while the network's routing algorithm reacts to the topology changes.

Unlike fixed infrastructure networks where link failures are comparatively rare events, the rate of link failure due to node mobility is the primary obstacle to routing in Ad-hoc networks. One possible solution for this problem, minimization of reaction to mobility, is to use a protocol based on a clustering system, this is the objective of this work, as well as to evaluate its advantages in terms of network performance measures. A protocol is a set of rules that governs the communications between nodes in a network. These rules include guidelines that regulate the following characteristics of a network: access method, allowed physical topologies, types of cabling, and speed of data transfer. A cluster is a group of similar things (e.g., nodes in telecommunications systems), with similar characteristics. Then, protocols based on a clustering system consist of a set of rules applying cluster partition, where nodes have autonomously organized themselves to form clusters. Each cluster contains a clusterhead, zero or more ordinary nodes and one or more gateways. For this work we are using the (α, t) clustering protocol in [3]. We describe these topics in following chapters.

Numerous challenges must be overcome to realize the practical benefits of ad hoc networking. These include effective routing, channel access, mobility management, security, and, quality of service (QoS) issues, mainly pertaining to delay and bandwidth management.

1.1 Objective

The purpose of this work is to evaluate the connectivity and quality as performance measures, using Trellis method to find paths and clustering algorithms in order to maintain an effective topology that adapts to node mobility so that routing can be more responsive and optimal when mobility rates are low and more efficient when they are high. The algorithm dynamically organizes the nodes of an Ad-Hoc network into clusters where probabilistic bounds can be maintained on the availability of paths to cluster destinations over a specified interval of time.

1.2 Justification

The main ideas of a communication system are to maintain an effective topology and provide service to the users at any time, therefore this service needs to be fast, efficient and of quality. To maintain an effective topology we can use a dynamical algorithm where nodes are organized into clusters where probabilistic bounds can be maintained on the availability of paths to cluster destinations over a specified interval of time.

In the case of Ad-Hoc networks, the main issues to be considered are the mobility, interference, connectivity and quality. When some problems related to these issues appear, it is necessary to have an organization within the network, through the nodes. This organization helps to diminish the negative effects of these issues and to maintain the acceptable connectivity and quality levels within the network.

1.3 Contribution

Nowadays in the Ad-Hoc network research, scientists have found methods to improve communications in this area. Some of them are several clustering algorithms, which present stable and unstable behavior, but at this moment and according to the literature, the most stable algorithms are based in clusters, one of these is Distributed Dynamic Clustering Algorithm (DDCA),[3], used in this work. Another topic in this thesis is to find k-paths in the network, to make this aspect we have several methods as Bellman Ford, Kruskal, Dijkstra, and Trellis, we are considering the last one to find connectivity and quality, easily, in the network,[19].

The major contributions of this thesis are the combination of clustering algorithms, in this case DDCA, and the use of the Trellis method to find k-paths. We also generate a stable Ad-Hoc network in a period of time and get performance measures as connectivity and Quality in the networks.

Another contribution is the variation in the Trellis Algorithm to find k-paths due to link costs because in this work we are using probabilities, these are explained in Chapter 3.

1.4 Organization

The organization of this thesis is as follows. In Chapter 2, the background on the study is introduced, a general description of Ad- Hoc networks and some important parameters that are mentioned throughout the thesis given. The Clustering processes and some works based in clustering for Ad-Hoc networks are also described. Chapter 3 contains clustering concepts in Ad-Hoc networks. In Chapter 4, we describe a model proposed. Chapter 5 shows the results using the model developed in chapter 4, and its analysis. Chapter 6 contains the conclusions of the thesis.

Chapter 2

Wireless Ad-Hoc Networks

A wireless Ad-Hoc network is a collection of mobile/semi-mobile nodes with no pre-established infrastructure, forming a temporary network. Laptop computers and personal digital assistants that communicate directly with each other are some examples of nodes in an ad-hoc network.

The term Ad-Hoc, tends to imply, "can take different forms" and can be mobile, standalone or networked. Nodes in the Ad-Hoc network are often mobile, but can also consist of stationary nodes; they should be able to detect the presence of other such devices and to perform the necessary handshaking to allow communications and the sharing of information and service. Since Ad hoc wireless devices can take different forms (for example, palmtop, laptop, Internet mobile phone, etc.), the computation, storage, and communications capabilities of such devices will vary tremendously. Ad-Hoc devices should not only detect the presence of connectivity with neighboring devices/nodes, but also identify what type the devices are and their corresponding attributes. These networks also have semi mobile nodes and they can be used to deploy relay points in areas where relay points might be needed temporarily, [1], [2].

Figure 2.1 shows a simple Ad-Hoc network with three nodes. Nodes A and C are not within transmitting range of each other. However, node B can be used to forward packets between them. Therefore, node B is acting as a router and the three nodes have formed an Ad-Hoc network.

An Ad-Hoc network uses no centralized administration, in other words an Ad-Hoc network is self-organizing and adaptive. This is to be sure that the network will not collapse just because one of the mobile nodes moves out of the transmitter range of the others. Nodes should be able to enter/leave the network as they wish.

Every node wishing to participate in ad-hoc network must be willing to forward packets for other nodes. Thus, each node acts both as a host and as a router. A node can be viewed as an abstract entity consisting of a router and a set of affiliated mobile hosts, in Figure 2.2 a router is an entity, which, among other things, runs a routing protocol.

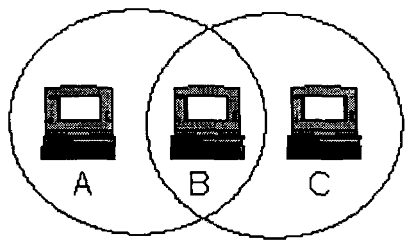


Figure 2.1: Example of a simple Ad-Hoc Network with three participating nodes

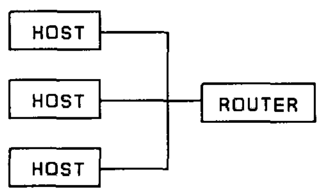


Figure 2.2: Block Diagram of a mobile node acting both as hosts and as router

Ad-Hoc networks are also capable of handling topology changes and malfunctions in nodes. It is fixed through network reconfiguration. For instance, if a node leaves the network and causes link breakages, affected nodes can easily request new routes and the problem will be solved. This will slightly increase the delay, but the network will still be operational, [2].

2.1 Important Aspects about Ad-Hoc Networks

In order to maintain good QoS in an Ad-Hoc Network, it is important that one know some main aspects about it, as Traffic Profiles, Types of Ad-Hoc networks, Security and privacy. We explain them in the following.

2.1.1 Traffic Profiles

Ad hoc wireless communications can occur in several different forms. For a pair of Ad-Hoc wireless nodes, communications will occur between them over a period of time until the session is finished or one of the nodes has moved away. This resembles a peer to peer communication scenario.

Another form occurs when two or more devices are communicating among themselves and they are migrating in groups. The traffic pattern is, therefore, one where communications occur over a longer period of time. This resembles the scenario of remote to remote communication. Finally, we can have a scenario where devices communicate in a non-coherent fashion and their communication session is, therefore, short, abrupt, and undeterministic.

2.1.2 Types of Ad-Hoc Networks

Mobile hosts in an Ad-Hoc mobile network can communicate with their immediate peers, that is, peer-to-peer, that are a single radio hop away. However, if three or more nodes are within range of each other (but not necessarily a single hop away from one another), then remote-to remote mobile node communications exist. Typically, remote-to-remote communications are associated with group migrations. Different types of Ad-Hoc communications result in different traffic characteristics, too.

2.1.3 Security and Privacy

Ad-Hoc Networks are intranets and they remain as intranets unless there is connectivity to the Internet. Such confined communications have already isolated attackers who are not local in the area. Note that this is not the case for wired and wireless-last hop users. Through neighbor identity authentication, a user can know if neighboring users are friendly or hostile. Information sent in an ad hoc route can be protected in some way but since multiple nodes are involved, the relaying of packets has to be authenticated by recognizing the originator of the packet and the flow ID or label.

2.2 Characteristics of MANETs

A MANET (Mobile Ad-Hoc Network) is defined as a collection of mobile platforms or nodes where each node is free to move about arbitrarily. Each node logically consists of a router that may have multiple hosts and that also may have multiple wireless communications devices, [3]. The term MANET describes distributed, mobile, wireless, multihop networks that operate without the benefit of any existing infrastructure except for the nodes themselves. Some characteristics about MANET are:

Dynamic Topology: Since nodes are free to move arbitrarily, the network topology may change randomly and rapidly at unpredictable times. The links may be unidirectional and bidirectional.

Bandwidth constrained, variable capacity links: Wireless links have significantly lower capacity than their hardwired counterparts. Also, due to multiple access, fading, noise, and interference conditions etc. the wireless links have low throughput.

Energy constrained operation: Some or all of the nodes in a MANET may rely on batteries. In this scenario, the most important system design criteria for optimization may be energy conservation.

Limited physical security: Mobile networks are generally more prone to physical security threats than are fixed cable networks. There is an increased possibility of eavesdropping, spoofing and denial-of-service attack in these networks.

2.3 Routing in Ad-Hoc Networks

Figure 2.3 depicts the peer level multihop representation of an ad-hoc network. Mobile node 1 communicates with node 2 directly, only one hop; otherwise, multihop communication is necessary where one or more intermediate nodes must act as a router between communication nodes. For example, there is no direct radio channel, between 1 and 3 or 1 and 5. Thus, nodes 2 and 4 must serve as an intermediate router for communication between 1 and 3 or 1 and 5, respectively. Indeed, a distinguishing feature of ad hoc networks is that all nodes must be able to function as routers on demand, [2].

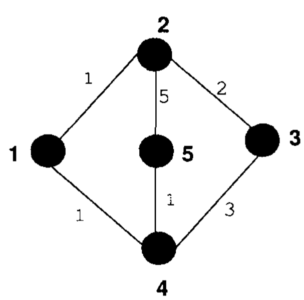


Figure 2.3: An Ad-Hoc Network example

An ad-hoc network begins with at least two nodes broadcasting their presence with their respective address information. Preferably, they may also include their location information, obtained using a Global Positioning System (GPS). If a node 1 is able to establish direct communication with node 2, verified by exchanging suitable control messages between them, they both update their routing tables. When a third node 3 joins the network with its beacon signal, two scenarios are possible.

The first is where both nodes 1 and 2 establish that single hop communication with 3 is possible. Second one is where only one of the nodes, in this case 2, recognizes the beacon signal from 3 and establishes the availability of direct communication with 3.

The distinct topology updates, consisting of both address and route updates, are made in all three nodes immediately afterward. In the first case all routes are directly. In the other, the route update first happens between 2 and 3, then between 2 and 1, and then again between nodes 2 and 3, confirming the mutual reachability between nodes 1 and 3 via 2. The mobility of nodes may cause the reachability relations to change in time, requiring route updates. Assume that for some reason the link between nodes 2 and 3 is no longer available. Nodes 1 and 3 are still reachable from each other, although this time only via node 4.

The network that we explained is a small network, but what happens if we have a large network?, we will probably have a serious problems with the mobility of each node, thus, we need schemes that work with this kind or problem, [2].

Existing schemes for routing in Ad-Hoc networks can be classified according to four broad categories, namely, proactive routing, flooding, reactive routing, and dynamic cluster-based routing. Figure 2.4 shows this classification. Proactive routing protocols periodically distribute routing information throughout the network in order to precompute paths to all possible destinations. Although this approach can ensure higher quality routes in a static topology, it does not scale well to large highly dynamic networks. By contrast flooding-based routing requires no knowledge of network topology. Packets are broadcast to all destinations with the expectation that they will eventually reach their intended target, [1].

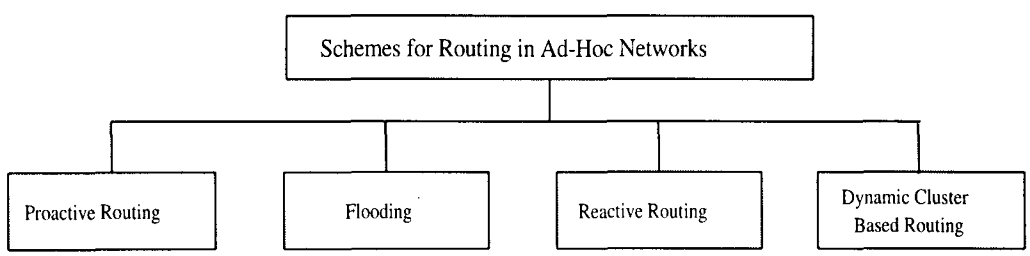


Figure 2.4: Schemes for routing in Ad-Hoc Networks

In reactive routing strategy, the design objective is accomplished by maintaining paths on a demand-basis using a query-response mechanism. This limits the total number of destinations to which routing information must be maintained, and consequently, the volume of control traffic required to achieve routing.

Other scheme is a dynamic cluster-based routing, in this scheme, the network is dynamically organized into partitions called clusters, with the objective of maintaining a relatively stable effective topology. The membership in each cluster changes over time in response to node mobility and is determined by the criteria specified in the clustering algorithm. In order to limit far-reaching reactions to topology dynamics, complete routing information is maintained only for intracluster (routing inside a cluster) routing.

Hierarchical routing has been shown to be essential in order to achieve at least adequate levels of performance in very large networks. In fixed infrastructure networks, hierarchical aggregation achieves the effect of making a large network appear much smaller from the perspective of the routing algorithm. The assignment of mobile nodes to cluster must be a dynamic process wherein the nodes are self-organizing and adaptable with respect to node mobility. Consequently, it is necessary to design an algorithm that dynamically implements the self-organizing procedures in addition to defining the criteria for building clusters.

The objective of the cluster framework is to maintain an effective topology that adapts to node mobility so that routing can be more responsive and optimal when mobility rates are low and more efficient when they are high. The algorithm dynamically organizes the nodes of an ad hoc network into cluster where probabilistic bounds can be maintained on the availability of paths to cluster destinations over a specified interval of time.

We need to consider that, large values of t tend to result in smaller clusters, whereas small values of t will increase the cluster size, which results in a better routing (optimal) with increased routing overhead, [2].

The cluster framework can also be used as the basis for the development of adaptive schemes for probabilistic QoS guarantees in ad hoc networks. Specifically, support for QoS in time-varying networks requires addressing:

1. Connection-level issues related to path establishment and management to ensure the existence of a connection between the source and the destination.

2. Packet-level performance issues in terms of the delay bounds, throughput, and acceptable error rates.

2.4 Ad-hoc Mobile Routing Protocols

In the previous section, we showed an example of an Ad-hoc Network, now we study some protocols applied for routing in this kind of network. Such protocols must deal with the typical limitations of these networks, which include high power consumption, low bandwidth, and high error rates. The following sections describe the protocols and categorize them according to their characteristics. Figure 2.5 shows a categorization of ad-hoc routing protocols, [4].

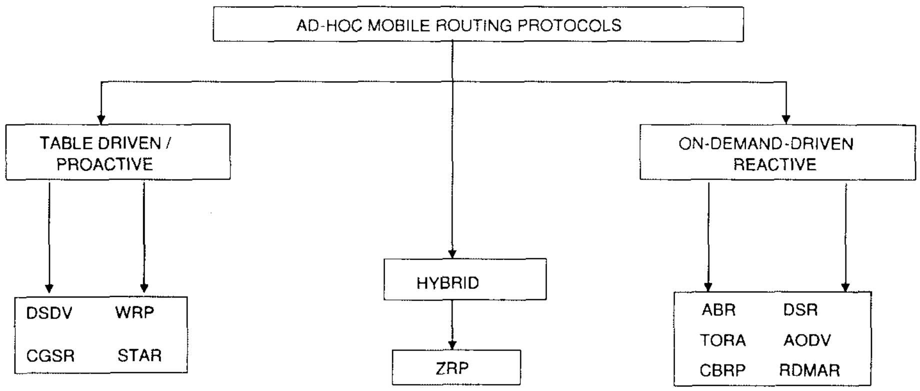


Figure 2.5: Categorization of Ad-Hoc Routing Protocols.

2.4.1 Table-Driven Approaches

Table-Routing protocols attempt to maintain consistent, up-to-date routing information from each node to every other node in the network. These protocols require each node to maintain one or more tables to store routing information, and they respond to changes in network topology by propagating route updates throughout the network to maintain a consistent network view. The areas where they differ are the number of necessary routing-related tables and the methods by which changes in network structure are broadcast.

Destination Sequenced Distance Vector (DSDV)

This routing protocol was developed at IBM, in 1996. The protocol is a distance vector protocol, which uses the modified Bellman-Ford algorithm. As said earlier, this is a table-driven protocol, where the route is always available. However, the protocol has some limitations as well. It maintains routing info among all the nodes, it uses periodic update messages, and there exists a route settling time and routes may not converge, [4], [7].

The protocol operates in the following way: mobile nodes maintain routes to all possible destinations and exchange routing info between each other. Hop counts are used as routing metrics, and in order to ensure that the routing information is up-to-date, sequence numbers are used. A given node keeps track of its own time and the sequence of events that happen. Thus the node assigns sequence numbers to distance vector updates, which updates contain information about the neighbors.

Cluster Switch Group Routing (CGSR)

This protocol was developed at UCLA in 1996. Some of the key features of the protocol are:

- it uses a clusterhead,
- code separation between the clusters
- cluster-based channel access and routing.

One limitation of this protocol is that it is based on DSDV as the underlying route update method, which can cause problems. The other limitation is that it uses periodic route and cluster membership updates, which result in additional overhead, [8].

The protocol is based on the concept of clusters and cluster-heads. Routing is done via the cluster-heads and gateways, as shown by Figure 2.6, where GW is a gateway node, $C1, C2$ and $C3$ are clusterhead to respective cluster, and $M1$ and $M2$ are origin and destine node, respectively.

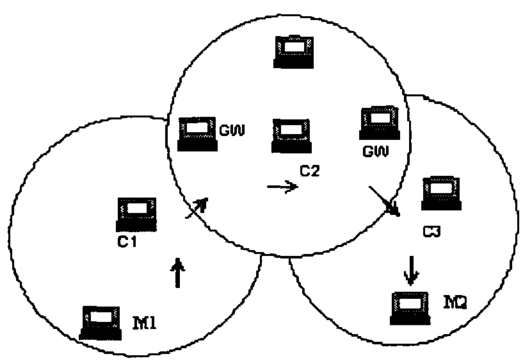


Figure 2.6: A CSGR path.

As show in Figure 2.6, a packet sent by a node is first routed to its clusterhead, and then the packet is routed from a clusterhead to a gateway to another clusterhead, and so on until the clusterhead of the destination node is reached. The packet is then transmitted to the destination.

Data from a host is routed in such a way that it is sent to the affiliated cluster-head, which then forwards that to a gateway node, which then sends it to the next clusterhead. The cluster member table is broadcasted periodically so that nodes can have an up-to-date information about the clusters. For this reason there is the need for cluster management.

There can be some problem regarding the routing efficiency in CSGR. If the mobile nodes use CDMA/TDMA, then it can take some time to get the permissions to send packets, as shown by Figure 2.7.

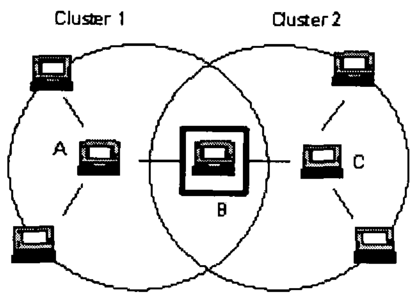


Figure 2.7: Routing Inefficiency in CGSR.

We have two clusters, with node A in Cluster 1 and node C in Cluster 2 and node B in both clusters, also as a gateway. If node A wants to send packets in Cluster 1, it must get permission to do so. At the same time node B, the gateway, must select the same code as node A to be able to receive the packet from node A. Then node B must select the same code as node C and get permission to send in Cluster 2.

As we said earlier, routing is done with the help of clusters and gateways. But one important issue is how to define a cluster-head in a cluster and how select such a clusterhead, so a cluster head election algorithm is used. The algorithm used is the Lowest ID or Highest Connectivity algorithm. This assumes that initially clusters are formed based on lowest ID or highest connectivity. If a ClusterHead (CH) moves from a cluster A to another cluster B, for example, then both clusters will give up its clusterhead appointment according to lowest ID or highest connectivity. In the same time, nodes detached from a cluster will recomputed their clustering according to lowest-ID or highest connectivity metric.

There are also some issues regarding cluster memberships, as cluster membership changes when cluster nodes migrate. Some interesting questions to solve are the following:

- how to derive unique cluster IDs
- how many nodes are allowable in a cluster.

One also must think about the fact that gateway nodes must be able to operate on two or more codes. All the above issues are non-trivial ones, and have to be taken into account when designing the protocol, [4].

Wireless Routing Protocol (WRP)

This protocol was developed at U.C. in Santa Cruz, in 1995. Just like for the other mentioned protocols, WRP simulations have been performed, but the protocol has never been implemented, [9].

WRP uses a distance-vector routing scheme, however it is a deviation from pure distance vector routing. In WRP, routes are always available. Some nice features of the protocol are that it is loop free and it also avoids the count-to-infinity problem by performing consistency checks of the reported predecessor information during routing updates. Some of the limitations of the protocol are the fact that it maintains routing information among all the nodes, thus it has to maintain many tables and it uses periodic update messages.

The update messages can be of two types: periodic route updates, which maintain routing info and event-triggered updates which are generated in response to mobility. The information maintained in a node is stored in a couple of tables, like the Distance Table (DT), Routing Table (RT), Link Cost Table (LCT) and the Message Retransmission List (MRL).

2.4.2 Source-Initiated On-Demand Approaches

An approach that is different from table-driven routing is source initiated on-demand routing. This type of routing creates routes only when desired by the source node. When a node requires a route to a destination, it initiates a route is found or all possible route permutations have been examined. Once a route has been discovered and established, it is maintained by some form of route maintenance procedure until either becomes inaccessible along every path from the source or the route is no longer desired, [4].

Associativity-Based Routing (ABR)

ABR was developed by C-K Toh at Cambridge University in 1996. The protocol is based on the concept of associativity, and new routing metrics are introduced, which are the longevity of a route (route relaying load) and the link capacity. The protocol is source-initiated thus there are no periodic route updates.

The basic idea of associativity is that there is no point in choosing a shortest-hop route if a route is going to be invalidated due to the node's mobility. Every node learns its 'association' with the surrounding nodes, where association can mean signal strength, power life, period of presence or spatial and temporal characteristics. The key idea is to choose a route that goes through nodes having a high degree of association stability, [4].

Dynamic Source Routing (DSR)

DSR was developed in 1996 at CMU. As it is suggested in the name of the protocol, it is based on the concept of source routing. The protocol uses the shortest path as routing metric. Routes are discovered on-demand, and caching is also used, [10].

The protocol operates in two phases: first route discovery is performed and then route maintenance. During route discovery first a route request is broadcasted to all the nodes, and then the route reply containing the route-record is propagated back to the source. During the route maintenance phase one has to take care of error packets and inform the source if necessary.

TORA - Temporally-Ordered Routing Algorithm

This protocol was developed at University of Maryland in 1996. It is claimed that it is implemented, but there are no experimental results reported, [11].

The protocol is on demand based; source initiated and uses the concept of link reversal. The concept used is based on building and maintenance of the 'height' metric, which aids routing. The 'height' is derived from the following values:

- logical time of link failure
- unique node ID
- ordering parameter
- reflective bit indicator

The limitations of the protocol are that it is timing dependent, information about adjacent nodes must be maintained and potential oscillations can occur. All this leads to a high protocol complexity. Due to its message passing nature it is suspected to have poor performance.

Similarly to the ABR protocol, TORA also has three phases of operation: the route discovery, route maintenance and route deletion. During the route discovery phase a Directed Acyclic Graph (DAG) has to be built from the destination. During route maintenance the DAG has to be rebuilt, while during route deletion a clear-packet is broadcasted.

AODV - Ad-hoc On-demand Distance Vector

The AODV protocol was proposed in 1997, after ABR and DSR appeared. It is claimed to be an enhancement over DSDV, but it is not really like that. The concept of distance vector is very weak, as it has changed from being table-driven to be on-demand. The implementation is pending, and also a multicast version is under evolution, [12], [13].

The protocol is essentially similar to DSR and it even has taken some part from ABR. Each node maintains its own sequence number. The protocol supports only symmetric links and may or may not use HELLO beacons. It operates by broadcasting RREQ messages until an intermediate node has a route to the destination or until it receives a reply from the destination itself. There is no route selection capability, as the node takes only the first route recorded by the first response on the RREQ. Nodes not in the selected route do not participate in route exchanges. Routes can be truncated, and in this case the source must be informed to redo the RREQ. Routes expire on soft state.

ZRP- Zone Routing Protocol

ZRP was developed at Cornell University in 1998 by Z. Haas and M.Pearlamn. The algorithm combines the proactive and reactive approaches and is built upon the concept of zones. So in this way ZRP uses table-driven routing for nodes within a routing zone (this is also called IntrAzone Routing Protocol - IARP) and on-demand query for nodes outside a routing zone (IntErzone Routing Protocol - IERP). Every node defines a zone radius, but the problem is that it is hard to decide what is an appropriate zone radius that is good for all applications, [14] , [15].

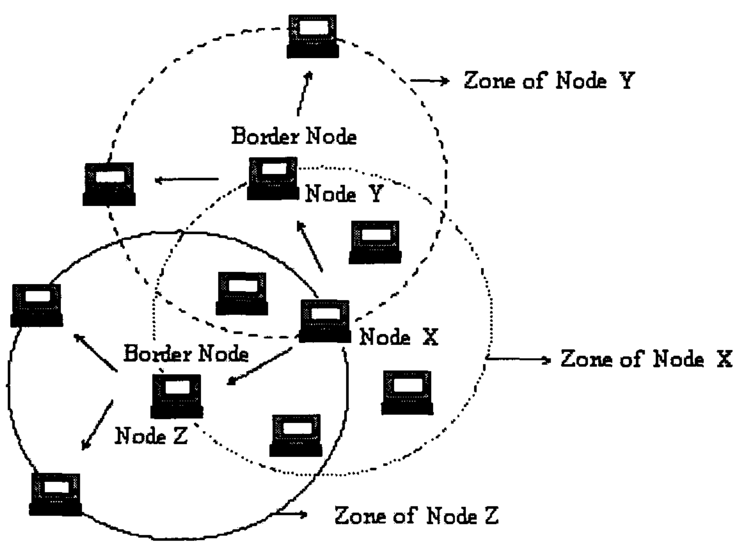


Figure 2.8: ZRP and Concept of Bordercasting.

Consider Figure 2.8, where it described the routing in ZRP, each node has routes to all other nodes in its own zone, which is achieved by using IARP. Routes to nodes outside a zone can be requested using IERP by broadcasting the request to all border nodes. This process is called bordercasting.

ZRP's IARP relies on an underlying neighbor discovery protocol to detect the presence and absence of neighboring nodes, and therefore, link connectivity to these nodes. Its main role is to ensure that each node within the zone has a consistent routing table that is up-to-date and reflects information on how to reach all other nodes in the zone.

IERP, however, relies on border nodes to perform on-demand routing to search for routing information to nodes residing outside its current zone. Instead of allowing the query broadcast to penetrate into nodes within other zones, the border nodes in other zones that receive this message will not propagate it further. IERP uses the bordercast resolution protocol.

Because parts of an ad hoc route are running different routing protocols, their characteristic will therefore be different. Some parts of the route is dependent on proper routing convergence, while the other part is dependent on how accurate the discovered interzone route is. This can make assurance of routing stability very difficult. Without proper query control, ZRP can actually perform worse than standard flooding-based protocols.

ZRP's route discovery process is, therefore, route table lookup and/or interzone route query search. When a route is broken due to node mobility, if the source of the mobility is within the zone, it will be treated like a link change event and an event-driven route updates used in proactive routing will inform all nodes in the zone. If the source of mobility is a result of the border node or other zone nodes, the route repair in the form of a route query search is performed, or in the worst case, the source node is informed of route failure, [4].

2.5 Quality of Service (QoS)

Quality of Service by the network is a guarantee to satisfy a set of predetermined service performance constraints for the user in terms of the end-to-end delay statistics, available bandwidth, probability of packet loss, and do on. The cost of transport and total network throughput may be included as parameters, [2].

The first essential task is to find a suitable path through the network, or route, between the source and destination that will have the necessary resource available to meet the QoS constraints for the desired service.

In Ad- Hoc Networks, we assume that each node carries a unique identity recognizable within the network. Following we assume the existence of all necessary basic capabilities, such as suitable protocols for medium access control and resource reservation, resource tracking, and sate updates. Each node periodically broadcast a beacon packet identifying it, thus allowing each node to learn of its adjacent neighbors.

Chapter 3

Clustering in Ad-Hoc Networks

In order to maintain an Ad-Hoc network organized, the network can be divided in partitions denominated clusters of nodes. A Cluster is a group of similar things (e.g., nodes in telecommunications systems), with similar characteristics. Then, protocol based on a cluster system consists of a set of rules applying cluster partition, where nodes have autonomously organized themselves to form clusters. Each cluster contains a clusterhead, zero or more ordinary nodes and one or more gateways. Clusterhead are nodes whose main functions are transmissions and allocation of resources within the clusters, for example, it might issue tokens to potential transmitters, emit busy tones when a transmission is in progress, or assign slots to specific transmitters and sessions. Gateways connect adjacent clusters. A gateway may directly connect two clusters by acting as a member of both, or it may directly connect two clusters by acting as a member of one and forming a link to a member of the other. Hence, the link-clustered architecture accommodates both overlapping and disjoints clusters, [5]. Figure 3.1 shows an example of a clustered network.

With the link-clustered architecture, all cluster members are within one hop of the clusterhead and hence within two hops of each other. This arrangement provides low delay paths between cluster members that may communicate frequently, and it places clusterheads in the ideal locations to coordinate transmissions among their cluster members. Clusterheads are distinct from gateways; hence, those for different clusters are separated by at least two hops. To establish a link-clustered control structure over a physical network, the nodes, [5],

- Discover neighbors to which they have bidirectional connectivity by broadcasting a list of those neighbors they can hear and receiving broadcasts from neighbors.

- Elect clusterheads and form clusters.

- Agree on gateways between clusters.

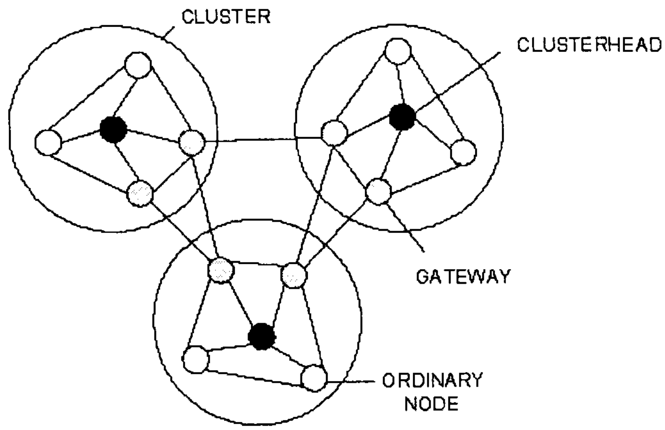


Figure 3.1: Example of cluster of Ad-Hoc network

3.1 Clustering Algorithm

The clustering algorithms to be discussed are completely distributed and adaptive. Hence, they are all suitable for highly dynamic wireless ad hoc networks. Distributed Mobility-Adaptive Clustering (DMAC) is based on the properties of the nodes, Distributed Dynamic Clustering Algorithm (DDCA) on the properties of the links, and Access-Based Clustering Protocol (ABCP) performs access-based clustering.

3.1.1 DMAC Algorithm

The Distributed Mobility- Adaptive Clustering (DMAC) algorithm determines the clusterheads not based on the node ID, but based on the nodes' generic weights, which is a positive real number. In fact, the DMAC algorithm does not depend on how the weight is computed. Ideally, the weight captures the mobility and reliability of a node and characterizes the preferences on which a node is suitable as clusterhead, [16].

Determining the clusterheads is performed fully distributed. Each node determines its role as either ordinary node or clusterhead on its own. Only local information is necessary. That is, the role of a node is determined by using its node ID and node weight, as well as the weights of all its neighbors and, for cases where ties occur, their node IDs. In order to allow for fast communication between two nodes, the DMAC algorithm requires every node to be connected to at least one clusterhead. Moreover, two clusterheads cannot be neighbors. This rule is used to ensure that the clusterheads are well spread out over the topology of the mobile stations.

The DMAC clustering algorithm works as follows. A node that is added to the network starts an initialization algorithm that determines it's role in the network, i.e. whether it should act as an ordinary node or a clusterhead. The decision is based on its own and its neighbors' node weight. If the new node has a neighboring clusterhead with a higher node weight, it decides to be an ordinary node, joins the cluster corresponding to that clusterhead and sends out a Join- message. Otherwise, it decides to become a clusterhead itself and sends out a ClusterHead- message, [16].

Thus, the neighbors get informed about the existence and role of a new node. The algorithm is message driven and works consistently if every node stores its own identifier, weight and role as well as the identifiers, weights and roles of all its neighboring nodes. In order to be adaptive to the dynamics of the wireless network, every node has to react on both Join and Clusterhead messages as well as changes in the surrounding topology. Possible changes are for example the appearance and failure of links or the appearance of new nodes.

The detection of those events is the responsibility of an underlying protocol. In conclusion, a stable, however temporary condition is reached if, first, every ordinary node is neighbor of at least one clusterhead, second, the affiliated clusterhead is the one with the highest weight, and third, two clusterheads are not neighbors. Figure 3.2 shows it algorithm.

3.1.2 ABCP Algorithm

An Access-Based Clustering Protocol (ABCP) designed for multi- hop wireless networks. Every ordinary node must be directly connected to the clusterhead. Basic design criteria were stable cluster structures and fast convergence, but moreover to keep the maintenance overhead as small as possible. The control channel plays an important role in this approach. The HELLO messages of the Medium Access Control (MAC) layer are included in the clustering process, [17], [18].

The clustering approach is access-based in the sense that a new node sends out a REQUEST TO JOIN message and joins the cluster whose clusterhead's HELLO message is received first. On receiving such a HELLO message the node sends out a JOIN message in order to inform its new clusterhead. If no HELLO message is received after a certain time, the node sends out a HELLO message instead and tries to become a clusterhead itself. A REQUEST TO JOIN message is also sent out if the link to the clusterhead gets weak or if the node receives a DISCONNECT message from its clusterhead. Both clusterheads and ordinary nodes have to send out a DISCONNECT message before they get inactive. Any REQUEST TO JOIN message is accepted by a clusterhead. This corresponds to the goal of maintaining the cluster structure.

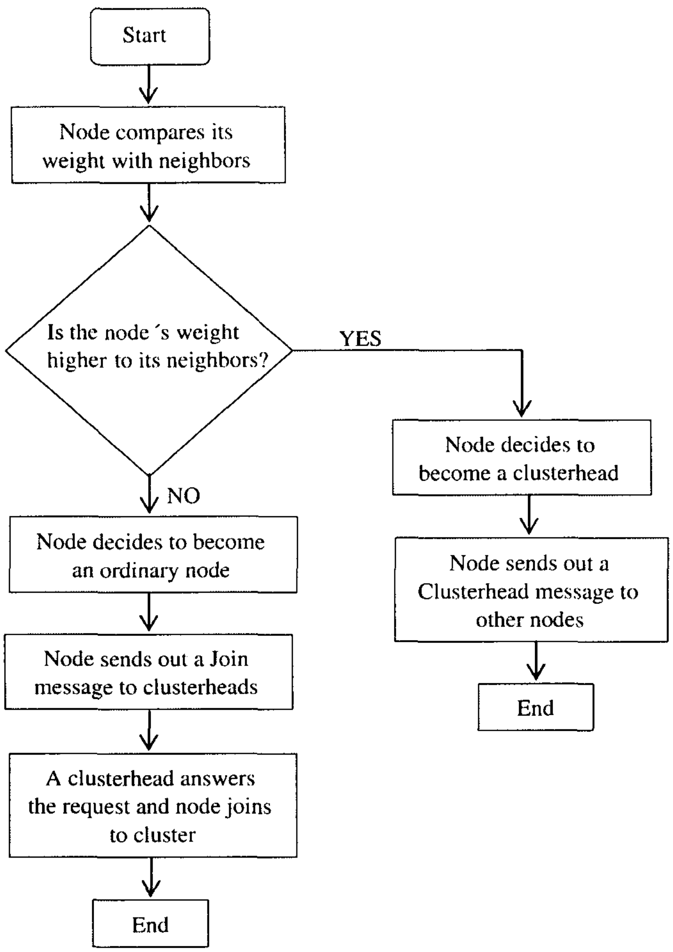


Figure 3.2: DCMA Flowchart

In order to maintain the original cluster as far as possible, clusterheads that are going to be inactive send out a SUCCESSOR message. This message declares the node with the highest number of direct links as new clusterhead, i.e., the one with the highest connectivity. Another refinement is that clusterheads that have no ordinary node in their cluster also send out a REQUEST TO JOIN message. If there exists at least one clusterhead this node is directly connected to, it will receive a HELLO message, send out a JOIN message and become an ordinary node. Figure 3.3 shows it algorithm.

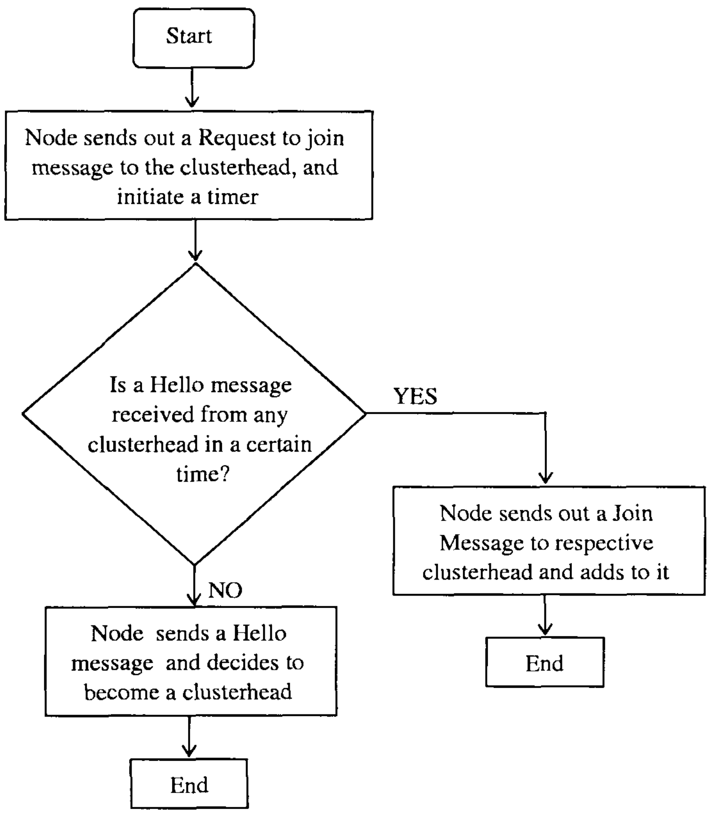


Figure 3.3: ABCP Flowchart

3.1.3 Application of the clustering algorithms to the sample scenario

In this section, the behavior of the approaches by the DCMA and ABCP algorithms in a sample scenario is described. A simplifying assumption made in the evaluation is that all sent packets are received correctly. In particular, the evolution of the clusters is investigated upon three basic changes in the topology: appearance of a new node, link failure and occurrence of a new link, [6], [16], [17], [18]. Figure 3.4 shows the initial network, where we have the link probabilities, identification and weight of each node in order to apply algorithm review before.

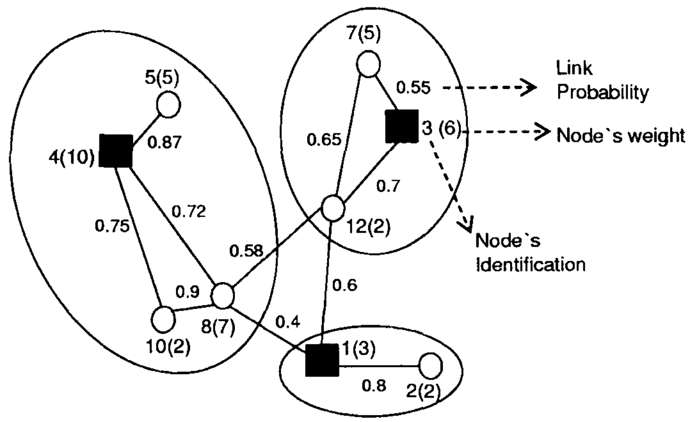


Figure 3.4: Initial Ad hoc Network

New Node

Figure 3.5 shows the sample network with an additional node 15 that was just activated. It has a node weight of 29 and is connected to both node 4 and node 5. Figure 3.5a shows the clustering result according to DMAC, Figure 3.5b according to ABCP. In the following, the reaction of the three presented clustering algorithms on this new node event will be described, [6], [16], [17], [18].

DMAC. The cluster formation completely changes because of the insertion of just one node, as shown in Figure 3.5a. The reason for that is that a new node does not accept a clusterhead with lower weight. Moreover, a chain reaction along nodes 15, 4, 8, 1, 2 occurs. The term chain reaction describes the fact that along a certain path, the roles of ordinary nodes and clusterheads invert. A chain reaction will always occur if (1), a new node has a higher weight than his clusterhead, (2), clusterheads and ordinary nodes appear alternately along a path, (3), the next node in the path has a lower weight than the node before, and (4), there is no ordinary node in the path having a clusterhead with higher weight than its predecessor in the path.

ABCP. According to the ABCP approach the new node simply joins the cluster of node 4. After sending out a REQUEST TO JOIN, node 15 receives a HELLO message from node 4. Note that even if link 15-4 did not exist, no severe changes in the clustering would

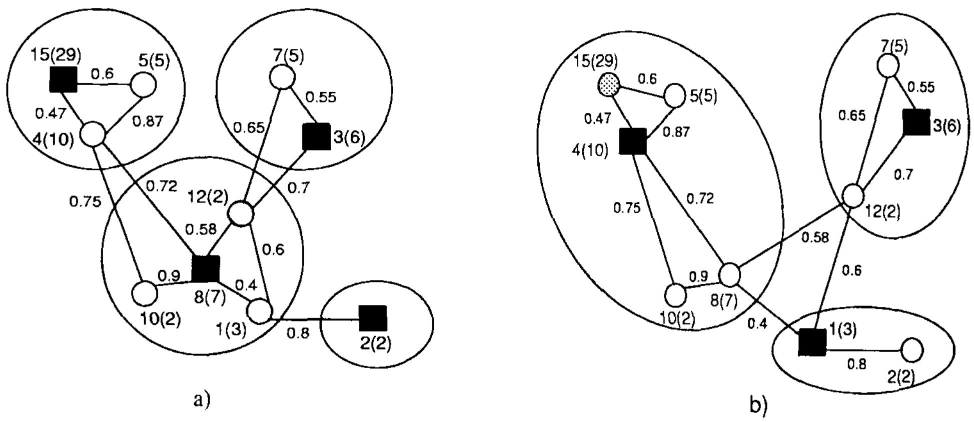


Figure 3.5: a) New Node, DMAC Algorithm. b)New Node, ABCP Algorithm.

occur. ABCP tends to form new clusters, changes are kept more locally and chain reactions don't occur. Hence, it is useful that in the refinements of ABCP, isolated clusterheads seek to join other clusters.

Link Failure

Lets consider the failure of the link between nodes 4 and 8. Figures 3.6a to 3.6b show the corresponding clustering by the different algorithms.

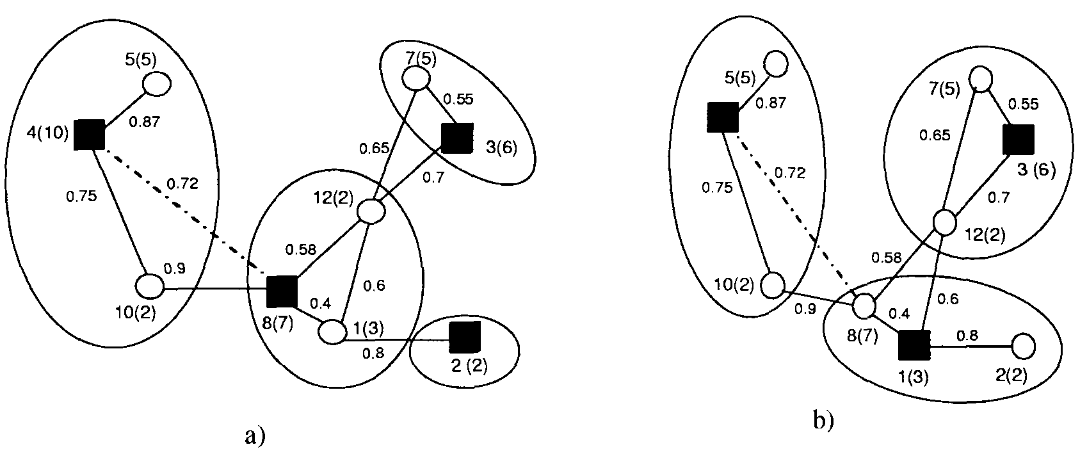


Figure 3.6: a) Link Failure, DMAC Algorithm. b)Link Failure, ABCP Algorithm.

DMAC. After the new node event, the re-clustering according to the new node event results in a severe change of the cluster structure, compare Figure 3.2a and Figure 3.3b. This happens even as there is only one link failure in an area with relatively high connectivity. However, this is due to the fact that a link between an ordinary node and a clusterhead failed. If, instead, a link between two ordinary nodes would fail, a stable clustering would be maintained.

ABCP. The only change in the cluster structure upon the link failure using ABCP is that node 8 now belongs to a different cluster, the one of node 1, as node 8 could receive node 1's HELLO message. The approach is obviously stable in cases where a second neighboring clusterhead exists.

New link

As one might expect, new link events due to the fact that mobile stations get connected that have not been connected before do not cause severe clustering changes. However, the DMAC algorithm does not allow two clusterheads to be neighbors. This is why in the following example it DMAC is the only approach that will change the cluster structure after a new link is available, see figures 3.7a and 3.7b, [6], [16], [17], [18].

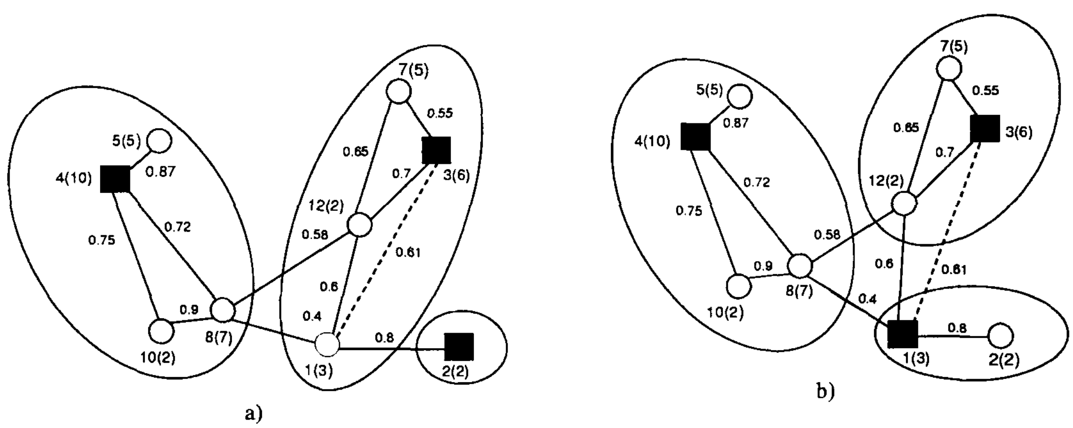


Figure 3.7: a) New Link, DMAC Algorithm. b) New Link, ABCP Algorithm.

3.2 DDCA Algorithm

The Distributed Dynamic Clustering Algorithm (DDCA) is claimed to provide both adaptive and robust clustering results over a wide range of mobility of the mobile nodes. This algorithm works with message driven and needs no periodic re-clustering but is continuously executed by all active nodes. The DDCA approach uses a so called (α, t) -criterion for clustering (edge based). The criterion describes a probabilistic bound, α, on the availability of paths in the corresponding cluster over a certain time t, [6]. Figure 3.8 shows a flowchart with DDCA algorithm.

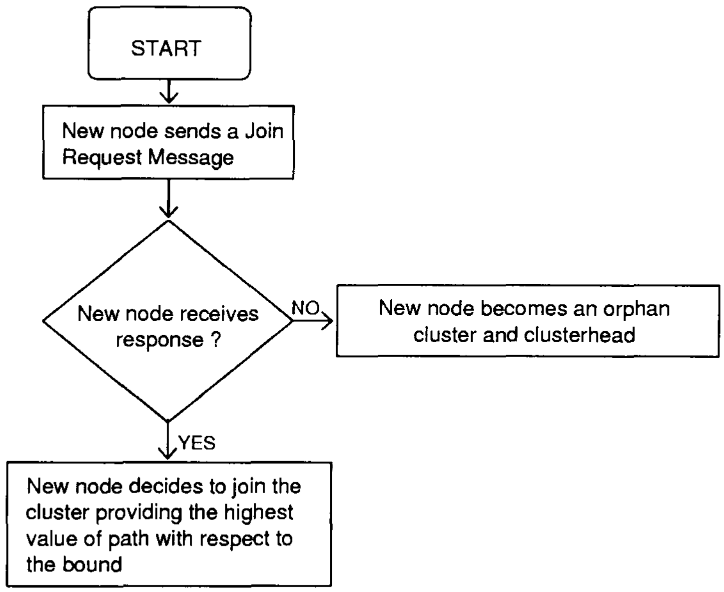


Figure 3.8: General flowchart for DDCA Algorithm

According to Figure 3.8 , a new node seeking a cluster to join sends out a JoinRequest message. If it does not receive any responses, it basically builds a new cluster and becomes clusterhead. This process is a little more involved as a deferral algorithm is used to handle situations with simultaneously broadcasted JoinRequest messages. If the new node upon sending a JoinRequest receives one or more JoinResponse messages before its JoinTimer runs out, it decides to join the cluster providing the highest (α, t)-value. This value has to exceed the minimum required value (α, t) threshold, otherwise the JoinResponse is ignored. The cluster strength, seen by a node m and given by the (α, t)- value, is a measure of the availability of a path from node m over some initial hop node n to the clusterhead of the corresponding cluster.

3.2.1 (α, t) Cluster Framework

Hierarchical routing has been shown to be essential in order to achieve at least adequate levels of performance in very large networks. In fixed infrastructure networks hierarchical aggregation achieves the effect of making a large network appear much smaller from the perspective of the routing algorithm. Cluster-based routing in ad-hoc networks can also make a large network appear smaller, but more importantly, it can make a highly dynamic topology appear much less dynamic. Unlike the cluster organization of a fixed network, the organization of an ad-hoc network cannot be achieved offline. The assignment of mobile nodes to clusters must be a dynamic process wherein the nodes are self-organizing and adaptable with respect to node mobility. Consequently, it is necessary to design an algorithm that dynamically implements the self-organizing procedures in addition to defining the criteria for building clusters, [3].

The objective of the (α, t) cluster framework is to maintain an effective topology that adapts to node mobility so that routing can be more responsive and optimal when mobility rates are low and more efficient when they are high. This is accomplished by a simple distributed clustering algorithm using a probability model for path availability as the basis for clustering decisions. The algorithm dynamically organizes the nodes of an Ad-Hoc network into clusters where probabilistic bounds can be maintained on the availability of paths to cluster destinations over a specified interval of time.

The (α, t) cluster framework can also be used as the basis for the development of adaptive schemes for probabilistic QoS guarantees in Ad-Hoc networks. Specifically, support for QoS in time-varying networks requires addressing:

- Connection-level issues related to path establishment and management to ensure the existence of a connection between the source and the destination.

- Packet-level performance issues in terms of delay bounds, throughput, and acceptable error rates.

3.2.2 (α, t) Cluster Characterization

The basic idea of the (α, t) cluster strategy is to partition the network into clusters of nodes that are mutually reachable along cluster internal paths that are expected to be available for a period of time t with a probability of at least. The union of the clusters in a network must cover all the nodes in the network, [3].

Definition 1: Let $P^k_{m,n}(t)$ indicate the status of path k from node n to node m at time t, $P^k_{m.n}(t) = 1$ if all the links in the path are active at time t , and $P^k_{m.n}(t) = 0$ if one or more links in the path are inactive at time t, Figure 3.9 shows this notation, here 1 and 2 are the possible paths from the origin node to the destination node, note that path 2 has an intermediate node r, while path 1, does not. The path availability $W^k_{m,n}(t)$ between two nodes n and m at time $t \geq t_0$ is given as, [3].

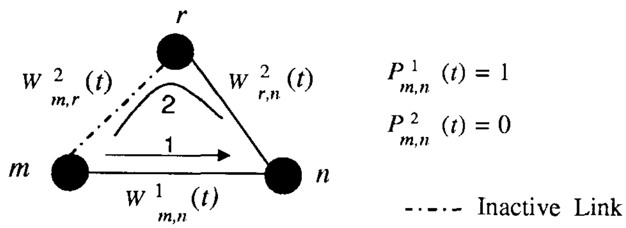


Figure 3.9: Notation to Definition 1.

$$W^k_{m,n}(t) \equiv Pr(P^k_{m,n}(t_0 + t) = 1) \mid P^k_{m,n}(t) = 1). \quad (3.1)$$

Definition 2: Let $W^k_{m,n}(t)$ be the path availability of path k from node n to node m at time t. Path k is defined as an (α,t) path if and only if,

$$W^k_{m,n} \geq \alpha. \quad (3.2)$$

Definition 3: Node n and node m are (α, t) available if they are mutually reachable over (α, t)paths.

Definition 4: An (α, t)cluster is a set of available nodes. Definition 4 states that every node in an (α, t)cluster has a path to every other node in the cluster that will be available at time $(t_0 + t)$ with a probability α. The cluster characterization, as previously defined, requires a model which quantifies the (α, t) path availability as given in Definition 1. Path availability is a random process that depends upon the mobility of the nodes which lie along a given path. Consequently, the mobility characteristics of the nodes play an important role in the characterization of this process.

3.2.3 Clustering Algorithm

Two key requirements motivate the design of a successful dynamic clustering algorithm, [3],

- The algorithm should achieve a stable cluster topology.
- It should do so with minimal communication overhead and computational complexity.

A parameter important in the Algorithm cluster is that every node in a cluster participates in a proactive routing protocol wherein the scope of routing information propagation is controlled by the nodes' view of their cluster membership. A node neither processes nor propagates routing information from nodes that do belong to its cluster is processed and disseminated. No centralized control over the clustering process is required. Nodes can asynchronously join, leave, or create cluster, [1].

In this Algorithm, we have four parameters to make a cluster:

- Node Activation
- Link Activation
- Link Failure
- Node Deactivation

We can describe these parameters in a flowchart as that of Figure 3.10, 3.11 and 3.12

Node Activation: The primary objective of an activating node is to discover an adjacent node and join its cluster. In order to accomplish this, it must be able to obtain topology information for the cluster from its neighbor and execute its routing algorithm to determine the (α, t) availability of all the destination nodes in that cluster.

The source node can join a cluster if and only if all the destinations are reachable via (α, t) paths. Such a cluster is referred to as a feasible cluster. The source node will continue checking each neighbor in sequence until it finds a feasible cluster or runs out of neighbors. If the source node is unable to join a cluster, it will create its own cluster, referred The cluster-join action is achieved asynchronously without any additional internodal coordination. The source node sets its node's Cluster Identifier Number (CID) to equal the CID of the cluster it is joining, and it generates its own routing update that is broadcast to its neighbors, [3].

Recognizing their own CID's in the routing update, those neighbors that are members of the target cluster process the source node's routing update. In doing so, the routing protocol automatically adds the source node as a destination in their respective routing tables, which infers cluster membership.

If the source node's network-interface layer protocol detects no adjacent nodes, or its attempts to join an adjacent cluster fail due to cluster infeasibility, the cluster algorithm generates and sets a globally unique CID that will be used in subsequent neighbor greeting exchanges. In this orphaned state, the (α, t)criteria is trivial because the path availability of the source node to itself is always one. In order to periodically reattempt to join a neighboring cluster, the node's timer is set to the value of the system parameter. Figure 3.10, shows this process.

Link Activation: A link activation detected by a clustered node that is not an orphan is treated as an intracluster routing event. The objective of an orphan node is to either have its own cluster expanded through the actions of other nodes or to join an existing cluster unless node mobility is very high. Link activation triggers an orphan node's attempt

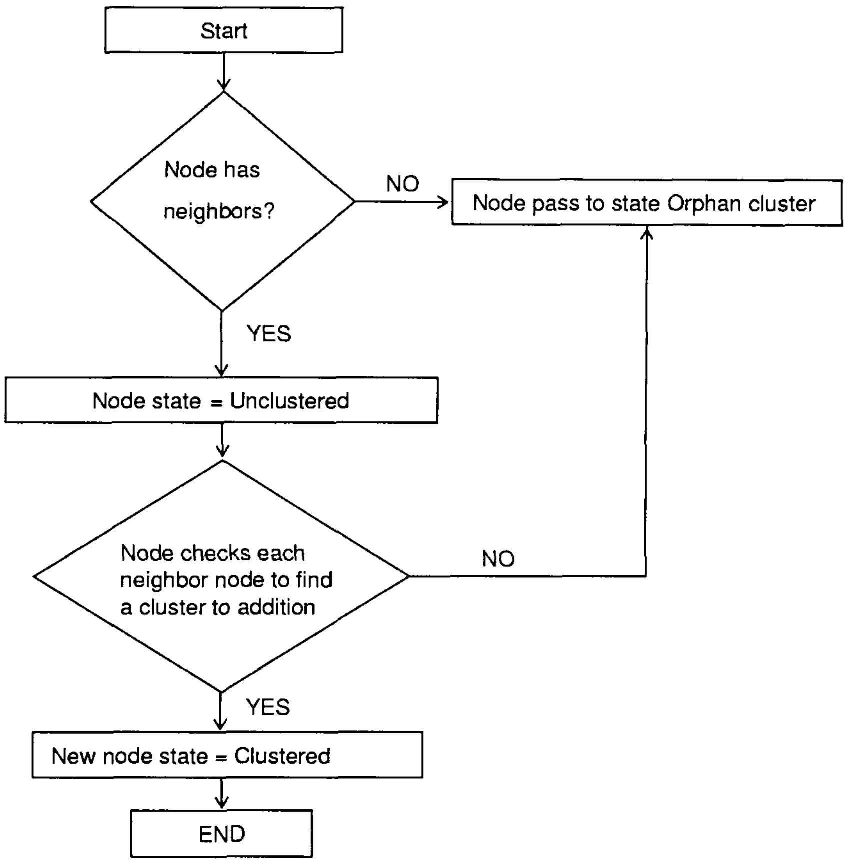


Figure 3.10: Node Activation Flowchart

to join a cluster. In order to receive cluster topology information from its new neighbor, the orphan node must temporarily reset its CID to indicate its unclustered status. Only information received from nodes that are in the same cluster as a destination or in the unclustered state are passed by the cluster algorithm protocol to the routing layer.

Thus, by changing its CID, the orphan node triggers the transmission of routing updates from its neighbor. Upon receiving the cluster topology information, the node evaluates cluster feasibility and either joins the cluster or returns to its orphan cluster status, depending upon the outcome of the evaluation. Figure 3.11, shows this process.

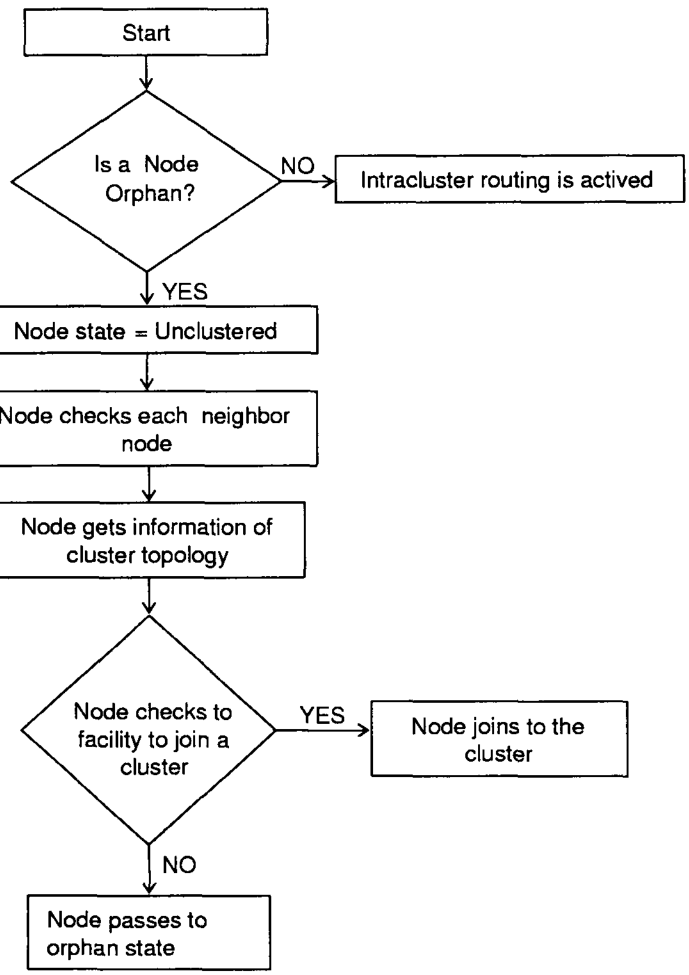


Figure 3.11: Link Activation Flowchart

Link Failure and Node Deactivation: In Figure 3.12, we show the process when a link failure or node deactivation. The objective of a node detecting a link failure is to determine if the link failure has caused the loss of any (α, t)paths to destinations in the cluster. A node's response to a link failure event is twofold. First, each node must update its view of the cluster topology and reevaluate the path availability to each of the cluster destinations remaining in the node's routing table. Second, each node forwards information regarding the link failure to the remaining cluster destinations. Each node receiving the topology update reevaluates its (α,t)paths as if it had directly experienced the link failure. When evaluating path availability to destination nodes within the cluster following a topology change, it is necessary to adjust the timing parameter to reflect that the timer has not yet expired. Use of the full value of would unnecessarily penalize the nodes by requiring a path availability that is higher (further out in time) than required by the cluster criteria. Thus, the estimated availabilities will reflect the probabilities evaluated at the maximum time for which this node has already made its probabilistic guarantee.

Using the topology information available at each node, the current link availability information is estimated, and maximum availability paths are calculated to each destination node in the cluster. If the node detects that a destination has become unreachable, then the node assumes that the destination has deactivated or otherwise departed from the cluster. In this case, the destination is removed from the node's routing table and will not be considered further in the evaluation of (α, t)paths. If a node detects that any of the remaining cluster nodes are connected within the cluster but not (α, t)reachable, it will voluntarily leave the cluster. A node leaves a cluster by sending a routing update to its neighbors that indicates that the status of all its links are down or equivalently an infinite distance to itself. It then resets its own CID to the unclustered value and proceeds according to the rules for node activation. No further action is required following a link failure if the node successfully evaluated (α, t)paths to each destination in the cluster,[3].

3.3 Trellis Algorithm for Ad-Hoc Networks

In order to analyze some concepts about quality and connectivity in Ad-Hoc networks, we need to find paths efficiently and easily to evaluate the performance measures. To use a Trellis algorithm is an alternative to pursue those objectives. In this case, we change the form to find the path cost, because we are using probabilities in the links cost; this change will be explained in the following sections. Now, we introduce some theoretical concepts and establish the notation and terminology, [19].

3.3.1 Graph Modeling

A directed graph $G = (V, E)$ is a structure consisting of a finite set of nodes $V = \{v_1, v_2, ...v_n\}$ and a finite set of links $E = \{(v_i, v_j) : v_i, v_j \in V \text{ and } v_i \neq v_j\}$, where

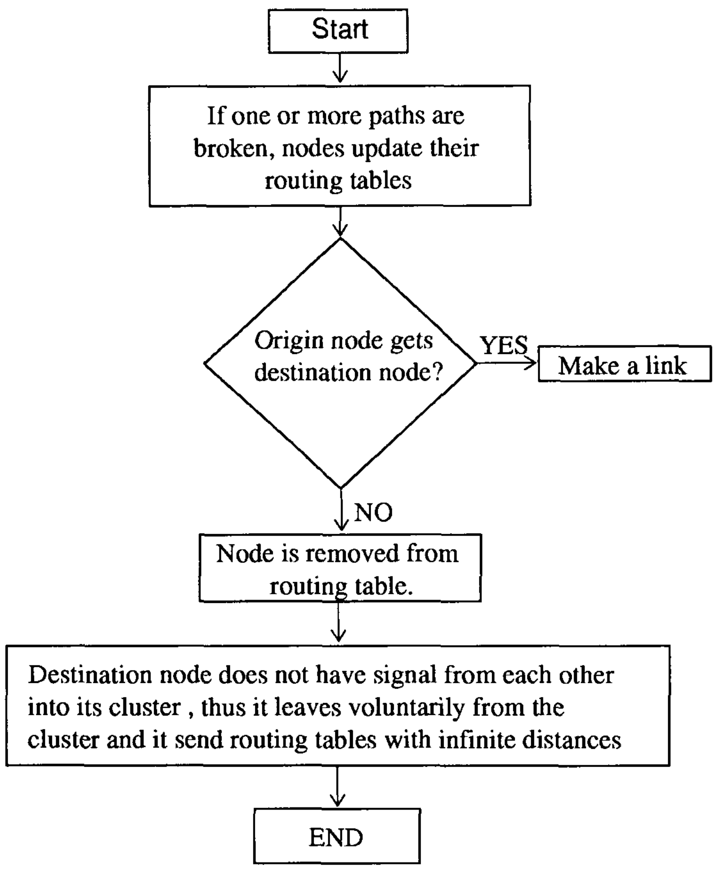


Figure 3.12: Link Failure and Node Deactivation

each link is an ordered pair.

Consider a node S (origin) and a destination node Z, we define a Trellis as a direct graph $G = (V, E)$ with nodes and directed links that satisfies the following conditions:

- The node set V s, z is partitioned into L subsets $V_1, V_2, .., V_L$ such that the cardinality of each set i is $\mid V_i \mid= H, 1 \leq i \leq L$, where H is the amount of nodes in each subset.
- Links connect nodes only of consecutive subsets V_l and V_{l+1} , for example, if $(v_i, v_j) \in E$, then $v_i \in V_l$ and $v_j \in V_{l+1}, i \leq z < L$.
- It has two more nodes $s \in V_0$ and $t \in V_{L+1}$ such that $(s, v_i) \in E$ for every $v_i \in V_L$ and $(v_j, z) \in E$ for every $v_i \in V_L, 1 \leq i, j \leq H$.

Figure 3.13 shows the parameters just described, where s is an origin node and z is a destination node. A walk on a trellis is an alternating sequence of nodes and links, i.e., $P = [v_1, (v_1, v_2), v_2, ..., (v_{k-1}, v_k), v_k]$. The length $L(P)$ of a walk is the path of links in it. A path is a walk in which all nodes are distinct.

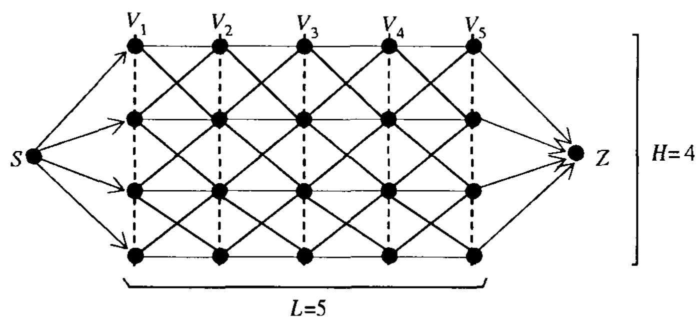


Figure 3.13: A K-Trellis graph with L=5, H=4

3.3.2 Link and Path Cost

Another parameter that needs to be studied in each link $(v_i, v_j) \in E$ of a trellis graph, $v_i \in V_l$ and $v_j \in V_{l+1}$, is the link cost and denoted $c(v_i, v_j)$, $1 \leq i, j \leq H$. Let $P = \{v_1, v_2, ...v_k\}$ be a path in a trellis graph. Consider $c(v_i, v_j)$ as the cost of $(i, j) \in E$, then the cost, $c(P)$, of a path P through the Trellis is defined as

$$c(P) = \sum_{v_i, v_j \in P} c(v_i, v_j), \tag{3.3}$$

The shortest path from node v_i to node v_j is a path $P = \{v_i, v_{i+1}, ...v_j\}$ with minimum cost.

For this work, as a contribution of this thesis, we are using probabilities as cost of links, therefore we change Equation (3.3) as

$$c(P) = \prod_{v_i, v_j \in P} c(v_i, v_j). \tag{3.4}$$

3.3.3 Algorithm Net to Trellis

Consider the network topology $G = (V, E, c)$, where c is the cost function from the link set E to a real number. The algorithm consists of the next steps, [19]:

1. The first step in the transformation process toward the trellis graph is to partition, with respect to a particular node, the node set of the network into adjacency- levels. By definition, this partition places the nodes of the network at vertical levels according to their distances.

2. In this step, we disconnect the network into two subnetworks G' and G''. Network G' contains all the nodes and links except the destination node t and the links incident to it. Network G'' contains the nodes in set$\{t\} \cup N(t)$ and all the links with end-nodes t and x, where $x \in N(t)$.

3. If now after step 1 and 2, we have any "vertical" links in G', for example two nodes connected by a link belonging to the same level, we apply two specific operations which eliminate "vertical" links. These operations are based on the addition of dummy nodes and 0-cost links, in such a way that the path cost is preserved.

4. In this step, we merge the resulting graph G' from step 3 with the graph G'' by adding (if necessary) dummy nodes and 0-cost links.

5. Now, if necessary, to complete the trellis graph, we introduce more dummy nodes, but with infinite link cost.

Definition: Let x, y be two nodes of consecutive levels which are connected by a link. The s-cost of link (x, y) is defined to be minimum cost of the path from s to y through node x , i.e., the cost of the path $P = \{s, ..., x, y\}$, and is denoted by $\phi(x, y)$

Operation P1: Let $G(V, E, c)$ be a network partitioned into adjacency-levels and let $(x, y) \in E$ be a link, where nodes x,y are on the same level l, i.e., $x, y \in AL(s, z)$, where $AL(s, t)$ are the adjacency-level sets. Figure 3.14a shows consecutive adjacency-levels of a partitioned network before and after the application of the Operation $P1$. Integer numbers indicate link cost, while integers in parenthesis indicate link s-cost. It is pointed out that before the application of the Operation $P1$ the link vertical (x, y) has only link cost, while after the operation it gains link s-cost.

We can see in Figure 3.14a that, links cost are positive integer numbers, and the link cost when a dummy node is presented is zero, but in this thesis we are using probabilities as links cost, therefore the link cost when a dummy node is presented is one (1) to no affect the path cost, explained in the section before. This is a contribution of this work.

Let x be the node satisfying the following properties

$$\min\{\phi(x,v) : v \in AL(s,l-1)\} \geq min\{\phi(y,u) : u \in AL(s,l-1)\},$$
$$\sum_{v\in N(x)\cap AL(s,l-1)} \phi(x,v) \geq \sum_{u\in N(x)\cap AL(s,l-1)} \phi(y,u).$$

Then, replace node x with a dummy node x', move node x into level $l+1$ and update the following parameters

$$w(x,x') = 0,$$
$$\phi(x,x") = \min\{\phi(x',v) : v \in AL(s,l-1)\},$$
$$\phi(x,y) = \min\{\phi(x,v) : v \in AL(s,l-1)\} + c(x,y).$$

Operation P2: Let $G(V, E, c)$ be a network and let x be a node at level l which, after operation $P1$, remains without neighborhoods in level $l-1$, i.e.,$N(x) \cap AL(s,l-1) = \emptyset$. Then, move node x into level $l+1$ and update the following parameter,

$$\phi(x,y) = \min\{\phi(x,v) : v \in AL(s,l-1)\} + c(x,y).$$

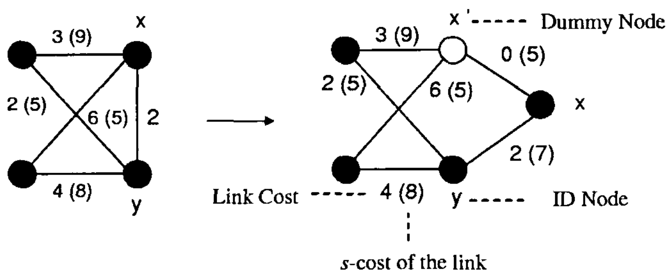
Figure 3.14b shows the resulting network topology when Operation $P2$ is applied to node x. Both links (z,x) and (y,x) have now link s cost.

In summary we have the formal listing to apply the algorithm described above.

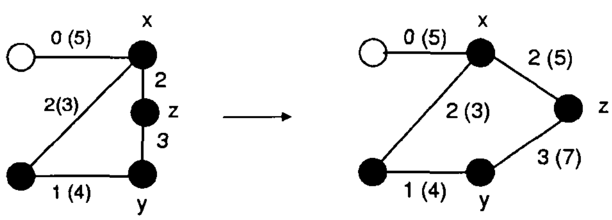
1. Partition the node set of the network $G(V, E, c)$ into adjacent-level sets, with respect to origin node $s \in V$. That is, the nodes of the network are placed at levels according to their distances from s; the origin s is placed at level 0.

2. Disconnected the network into two subnetworks G' and G'', where G'contains all the nodes and links except the destination node t and the links incident to it, while G'' contains the nodes in set $\{z\} \cup N(z)$ and all the links of the form (x,z), where $x \in N(z)$.

3. Apply the operations $P1$ and $P2$ in the network G' in order to eliminate all the "vertical" links of the network G'.



a) Illustration of Application of the Operation $P1$



b) Illustration of Application of the Operation $P2$

Figure 3.14: Illustration of application of the Operation $P1$ and $P2$.

4. Merge the resulting network from step 3 with the network G'' by adding dummy nodes and 0-cost links.

5. Complete the trellis structure by adding ∞ -cost links.

Figure 3.15 shows the flowchart about Trellis Algorithm.

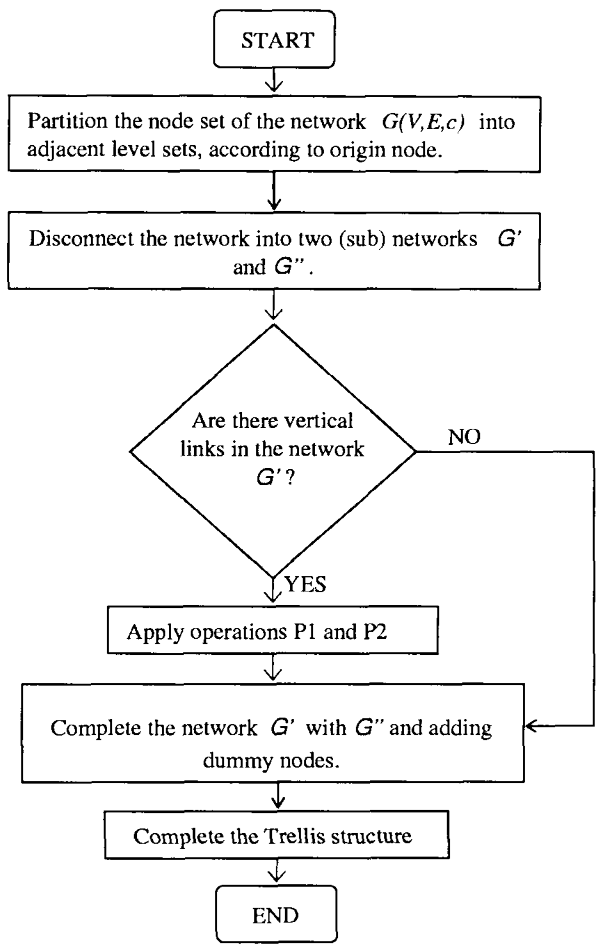


Figure 3.15: Trellis Algorithm Flowchart

3.3.4 Example applying Trellis Algorithm

For this work, we apply Equation (3.4) due to link costs are probabilities, Figure 3.16 shows it.

According to section 3.3.3, where dummy links had a zero value due to they are not probabilities, but in this case dummy links have one-value because we are using probabilities as link costs. See Figure 3.16, where p_1, p_2, p_3 are link probabilities.

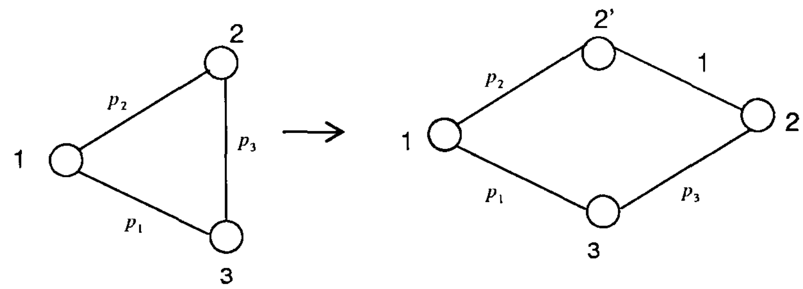


Figure 3.16: Transformation from normal network to Trellis Graph

In Figure 3.16 we observe the transformation form a normal Network to Trellis Graph. For this transformation, we apply the steps described before in this chapter, but applying the new parameters for dummy link and link cost. For example if we consider network in Figure 3.17, and we try to convert it to a Trellis Graph to find the k-paths. Figures 3.16, 3.18, 3.19 and 3.20 show these.

If we take each cluster and apply the Trellis Algorithm we obtain an easily form to find k-paths into each cluster, and after that we take each cluster as a node and apply the algorithm again.

According to Figure 3.18 in a) Step 1 and 2 from the Trellis algorithm to Cluster A, b) Step 3 and 4 from Trellis algorithm to Cluster A, c) Step 5 from the Trellis algorithm to Cluster A. Now, we are applying this method in all cluster, separated, we can see Figure 3.19 for cluster B, and Figure 3.20 for cluster C.

Now, we need to know paths between clusters, so we apply again the trellis algorithm, but now we take the border nodes in the cluster as nodes in the network, in Figure 3.21 we can see that.

Now, applying Equation (3.4), we can obtain all possible paths from origin to destination, according to the procedure developed before. Figure 3.22 we can see it.

Figure 3.22 shows the best path from node 5 to node 3 and pass trough nodes 4, 8, 12, according to equation (3.4), where,

$$c(P) = (0.87)(1.00)(0.7)(0.5)(1.00)(0.7)(1.00) = 0.21924.$$

Figure 3.23 shows other paths between node 5 to node 3 with minor probabilities, the following result show the probabilities for each path.

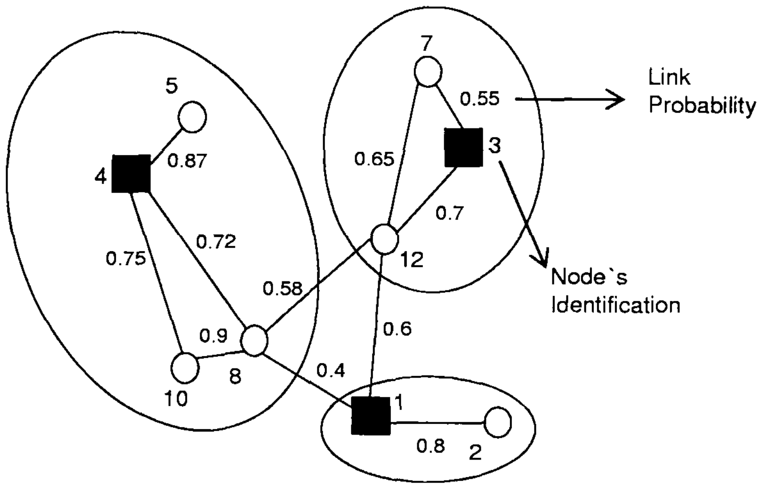


Figure 3.17: Initial Ad-Hoc Network.

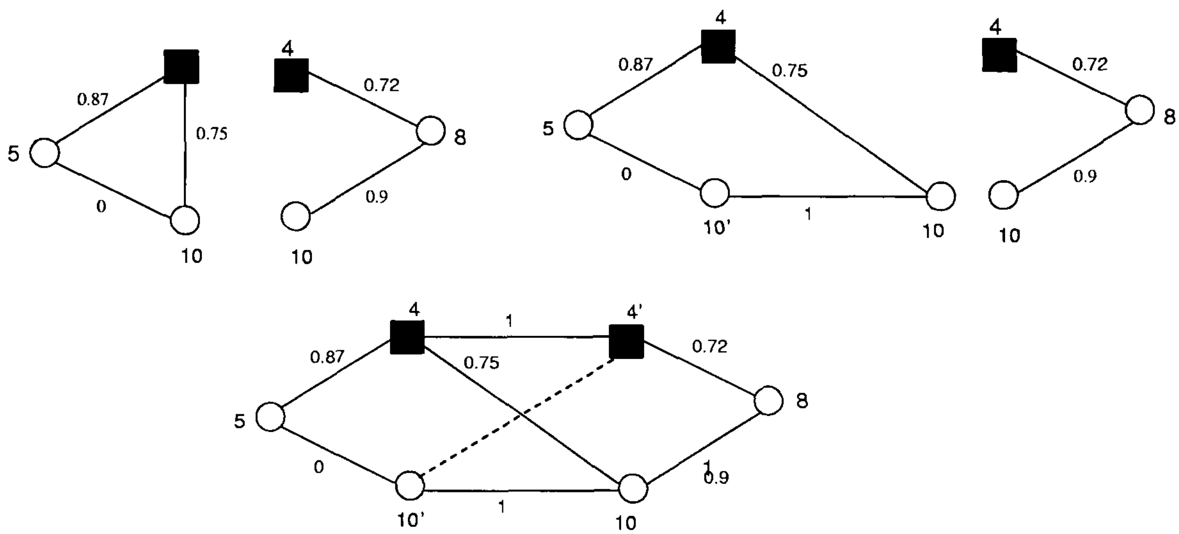


Figure 3.18: Applying the Trellis Algorithm in Cluster A.

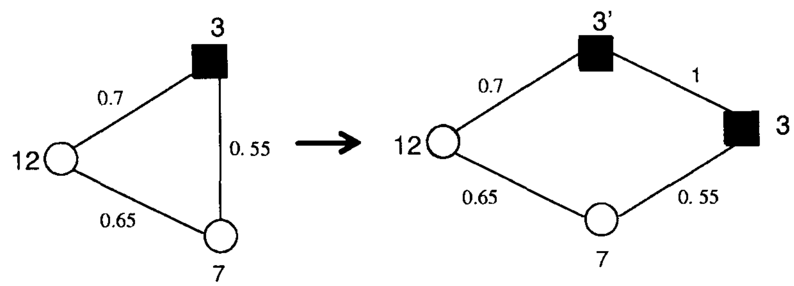


Figure 3.19: Trellis Algorithm applied to cluster B

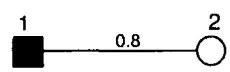


Figure 3.20: Trellis Algorithm applied into cluster C

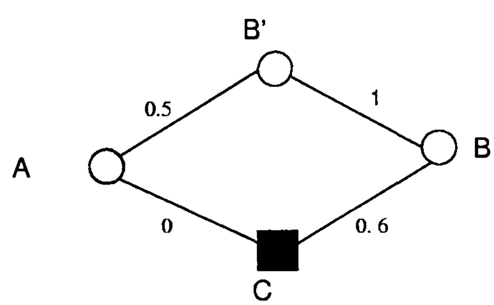


Figure 3.21: Trellis Algorithm applied taking clusters as a nodes

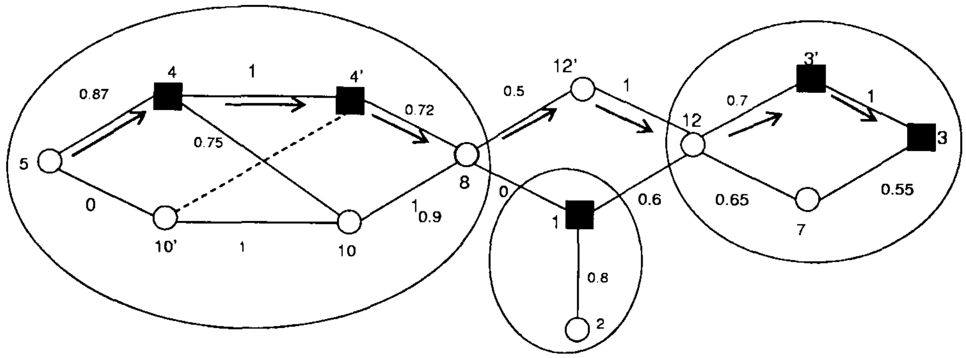


Figure 3.22: Trellis Algorithm applied in all the network

For path a), we have $c(P) = (0.87)(1.00)(0.75)(0.9)(0.5)(1.00)(0.7) = 0.20553$, this path is compose by nodes 5,4,10,8,12,3.

For path b), we have $c(P) = (0.87)(1.00)(0.72)(0.5)(1.00)(0.65)(0.55) = 0.11196$, this path is compose by nodes 5,4,8,12,7,3.

For path c), we have $c(P) = (0.87)(1.00)(0.75)(0.9)(0.5)(0.65)(0.55) = 0.10497$, this path is compose by nodes 5,4,10,8,12,7,3.

Now we have three scenarios,

- Link Failure
- New Link
- New Node

Link Failure: In this case , if for example link between nodes 12 and 3 fail, then we need to change the path, so, we are taking the path with higher probability, see, figure 3.24.

Then, the best option to take a path is path b) where $c(P) = 0.11196$.

New Node : If new node is added, only is affected the cluster to it belonging for example, taking Figure 3.5b and apply trellis Algorithm we obtain Figure 3.25. Note that the paths are the same to communicate between nodes 5 to 3.

New link: If new link is added, for example in Figure 3.7b, we analyze only the cluster o clusters affected. We can see this scenario in Figure 3.26.

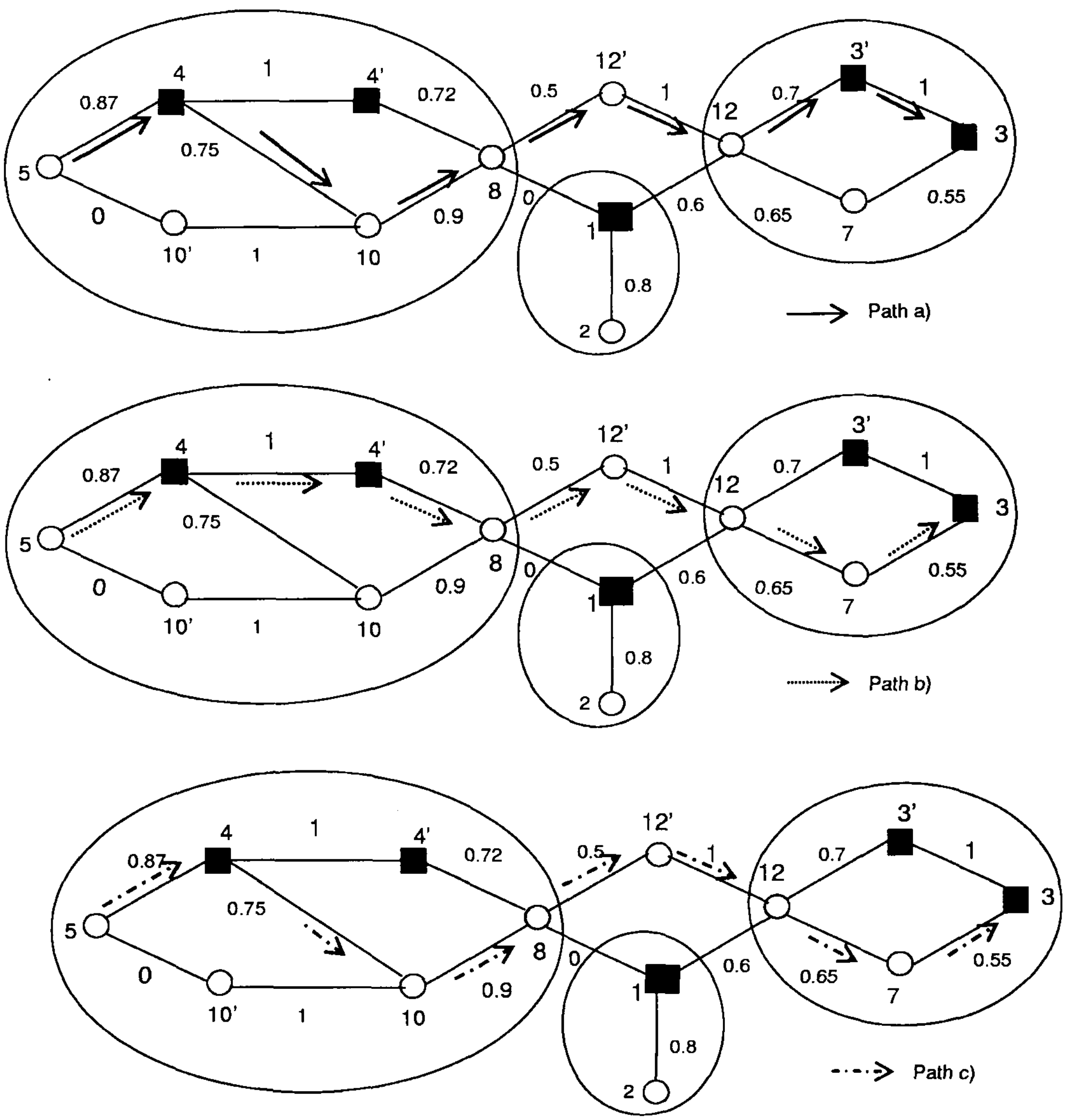


Figure 3.23: Paths with minor probabilities.

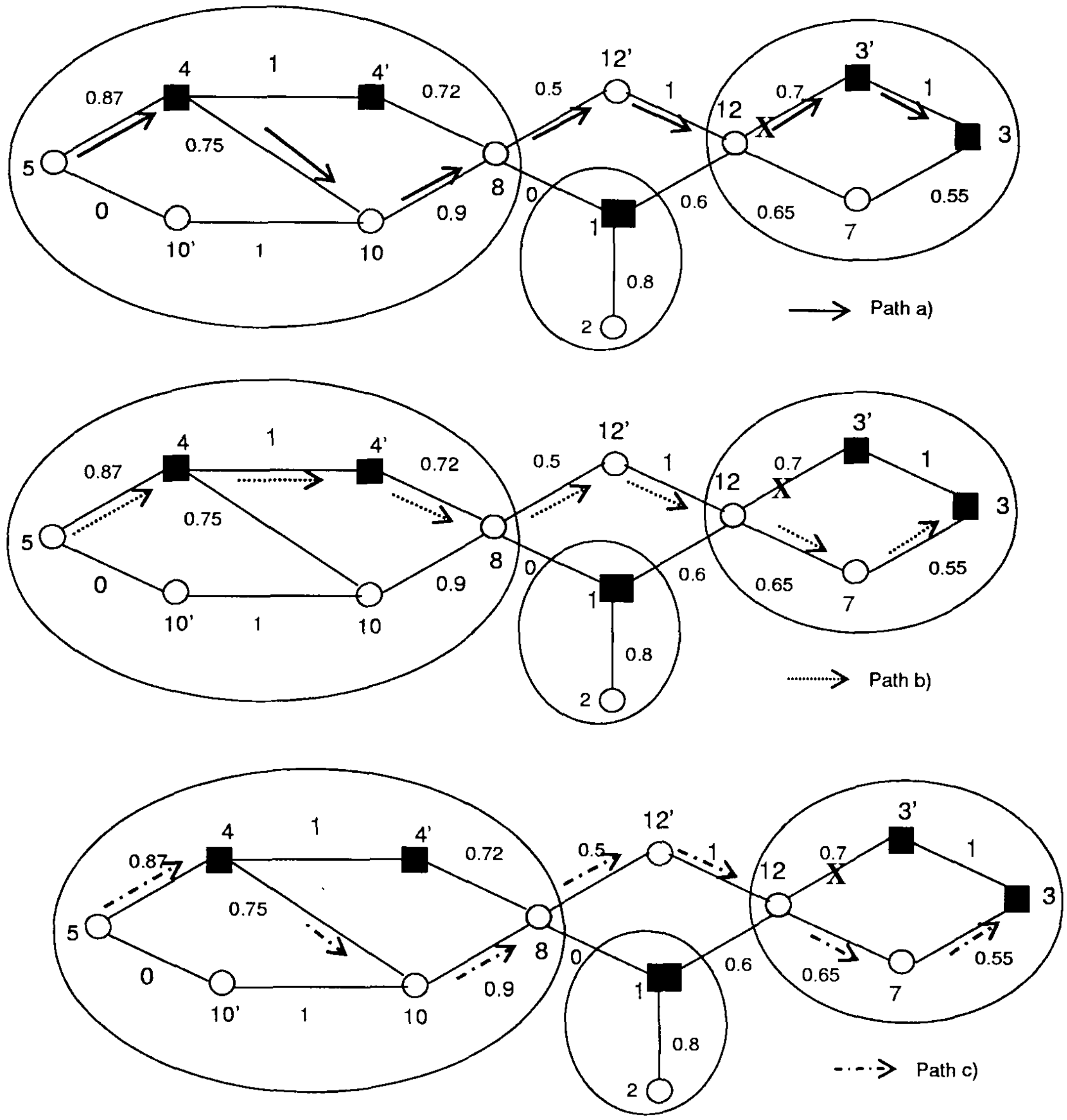


Figure 3.24: Link Failure Scenario.

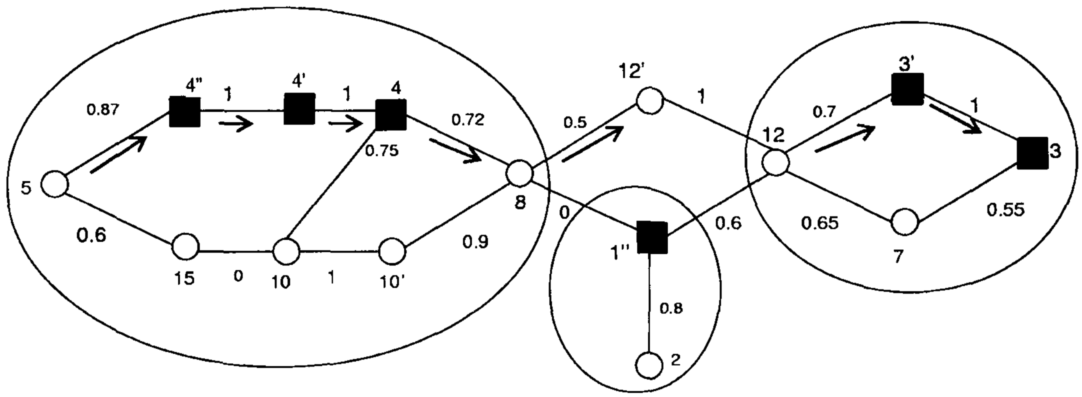


Figure 3.25: New Node Scenario.

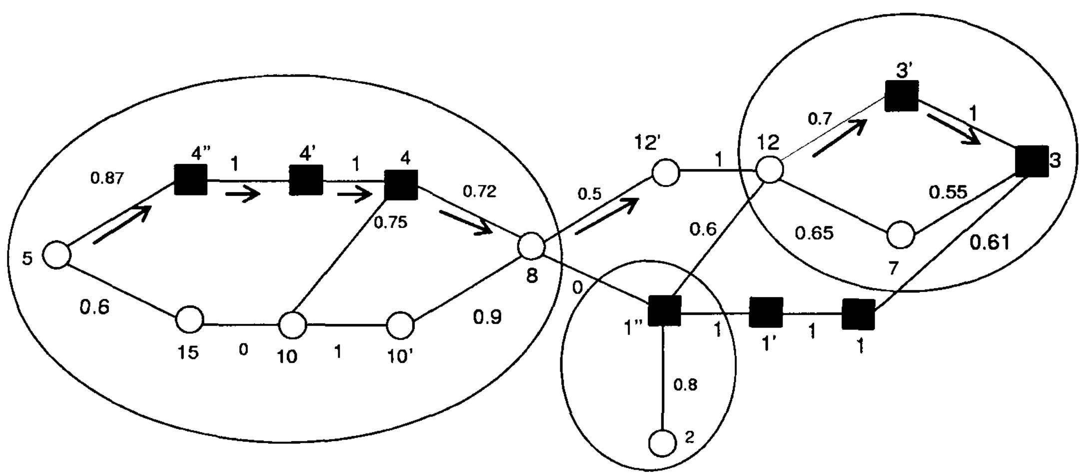


Figure 3.26: New link Scenario.

The Trellis Algorithm is more difficult to implement than others (as Bellman Ford, Dijkstra), because with this algorithm we find all the possible paths between two nodes (origin, destine), while in the others we can find only the path with major cost. Therefore, in case when a link fail in a path chose for communication, for example, it is not necessary to recalculate the paths again because they (origin, destine) have the other options of the paths calculated before. So, the response time, when these kind of cases occur, is minimum compared with other algorithms where is necessary calculate the path again.

3.4 Connectivity and Quality of Network

The objective for this thesis is to evaluate two important parameters in an Ad-Hoc network as Connectivity and Quality of the network. Now, we provide some basic concepts about them.

Connectivity is nodes' ability to communicate and share information with other nodes. See Figure 3.27. According to this definition we can build a complex topology in the network with several nodes and paths between them. To facilitate the analysis of the network, we are using clusters as nodes where each cluster can be communicated with others (Connectivity definition), therefore the network topology is dynamic and easy to study. In our case, this connectivity is a function of the distance between nodes, where the probability to communicate increases when the distance between a pair of nodes decreases compared to the coverage radius of the nodes, and viceversa. Connectivity is based on the number of nodes that are reachable in the network and within the clusters.

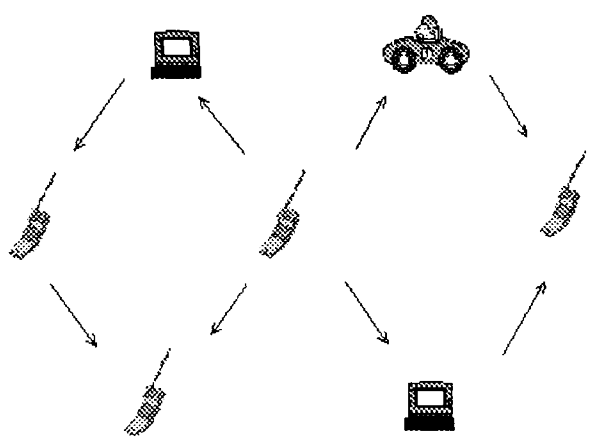


Figure 3.27: Connectivity example.

In Ad-Hoc network, when the density of nodes per unit area is high, interferences and contentions for medium access limit their capacity. When the density is low, interferences are then less critical, contrary to connectivity. See Figure 3.28

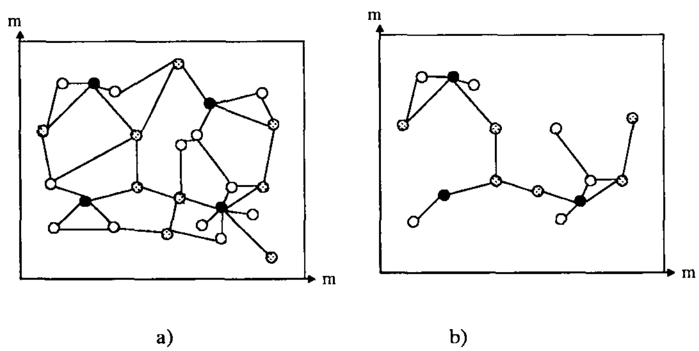


Figure 3.28: a) High Density of nodes, b) Low density of Nodes.

Quality is the form to guarantee certain transmission characteristics node to node in the network. This implies a concept about Quality of service whose objective is to provide better network services in order to satisfy customer network applications. In Chapter 2 we have thorough definitions about it. In our case, this Quality is based on the distance between nodes, and the corresponding link probabilities with higher quality for higher probabilities.

Chapter 4

Model Description

In this thesis we are interested in presenting a mathematical model to analyze the process of clustering in wireless Ad-Hoc networks and evaluate its advantages in terms of network performance measures.

4.1 Network Generation

To apply the algorithm described in Chapter 3, we need to consider some important parameters as cluster partition, area, radius of the cluster, clusterhead, and border nodes. In the next sections we describe the algorithm proposed to generate the parameters mentioned in a general form.

1. The first step is to generate the nodes. We use a Poisson distribution of users in the network.

2. Determination of area, number of clusters and radio of the clusters. To do this step, we define a square area and a square form for the clusters. Taking a number of clusters,NC, with exact square root, e.g., 4 , 9, 16, we can cover all the area and find easily the number of clusters per side, NCS, with

$$NCS = \sqrt{NC}, \tag{4.1}$$

3. Determination of coordinates to central points into the clusters.

4. Find nodes belonging to each cluster, depending on their position in the area.

5. To find the clusterhead in each cluster, we need to calculate the Euclidian distance, d_{ij}, between the cluster's central points of step 3, in this case let the coordinate (h, k)

be that of node i, and let the other node j belonging to the cluster, have coordinates (x, y), then we have

$$d_{ij} = \sqrt{(x-h)^2 + (y-k)^2}, \tag{4.2}$$

6. Knowing the distances,d_{ij}, we determine the real clusterhead (between central cluster points, (h, k) coordinate, and their nodes belonging to its cluster, (x, y) coordinate) and taking the smallest value among them. This value corresponds to the nearest node to the central point in the cluster, therefore this node will be clusterhead. We repeat this step in each cluster in the network.

7. To determine the border nodes and the links between nodes, we take a coverage area of the clusterhead node as a circle, to have the same coverage radius, R, around it.

8. To establish the border nodes, we calculate the distance, d_{ij}, from the clusterhead i to each node j within the cluster, using (4.2) and take the maximum values according to the coverage radio of the clusterhead. The number of border nodes depends on the number of nodes in the cluster, since not all of them can be border nodes.

9. Next step is to calculate the probabilities of the links, called link probabilities, of the network between nodes. To obtain these probabilities, we take the coverage radius, R, and the Euclidian Distance, d_{ij} calculated in step 8, according to Figure 4.1, and obtain, [20],

$$P(link) = 1 - F_d(d), \tag{4.3}$$

   where,

$$F_d(d) = (\tfrac{d_{ij}}{R})^2,$$

   represent the probability of communication from a node to another one and depends on the distance, in this case if d_{ij} is equal to R, the probability is one, but in Ad-Hoc networks, if we have a greater distance from two nodes the communications can be difficult, therefore, we taking the complement to find the link probabilities.

10. Now we can reach nodes that are not in to the clusterhead coverage area, but are in the cluster, to make this, we apply the Trellis Algorithm and find all the paths from clusterhead (maximum two hops) to nodes outside from clusterhead coverage area, by Equation (3.4) and choose the path with major value, according to value of α,

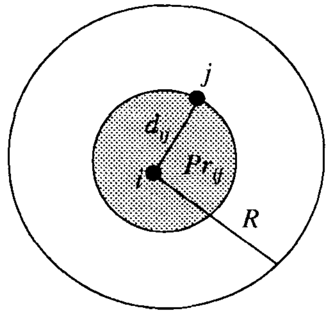


Figure 4.1: Illustration of nodes, distances and Pr_{ij} of (4.3)

that is the bound establish as a threshold for link probabilities.

11. Assigned value to α parameter and compare this with probabilities per link, if

$$\alpha > P(link) \text{ then } P(link) = 0,$$
$$\alpha \leq P(link) \text{ then } P(link) = \text{Calculated value.}$$

12. Now, the network is ready and we can apply the concepts of Connectivity and Quality that will be described in the next section.

Figure 4.2 shows, the algorithm proposed to Network Generation.

4.2 Connectivity and Quality Application

Next step is the application of Connectivity and Quality concepts to the network generated. Analyzing the network we find various forms to apply the concept of connectivity, Figure 4.3 shows them.

Connectivity into each cluster is the connectivity related to the amount of reachable nodes into each cluster. A reachable node is a node that can be attainable by one or two hops from the corresponding clusterhead. Let $C.Cluster$ be this Connectivity into each cluster, AN be the number of reachable nodes and NN be the total number of nodes in the cluster, then the connectivity into each cluster is given by

$$C.Cluster = \frac{AN}{NN}, \tag{4.4}$$

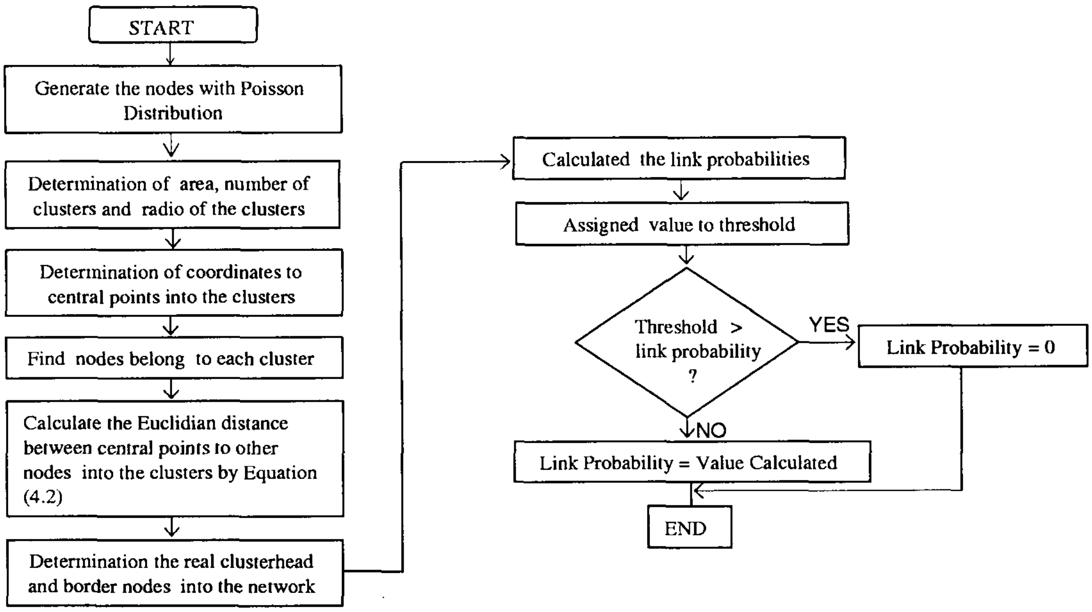


Figure 4.2: Network Generation Algorithm

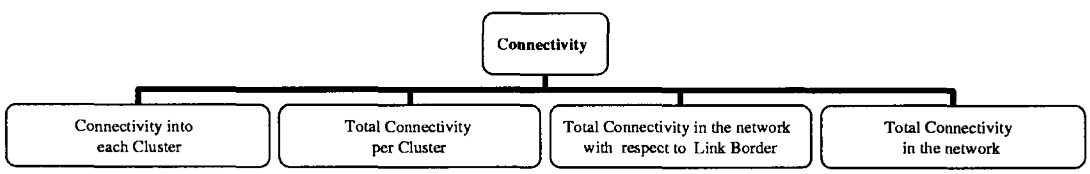


Figure 4.3: Types of Connectivity at the network generated

Connectivity per cluster is the connectivity per cluster respect to the whole network. Let $TCCluster_i$ be the Total Connectivity per each cluster i, $\overline{C.Cluster_{ij}}$ be the average of $C.Cluster$ between two clusters, cluster i and cluster j, where connectivity of cluster i is analyzed; these clusters must have an availability link between them. We consider two clusters to be adjacent whenever they have at least a link with positive cost between them.

Let NL_i be the total number of clusters adjacent to cluster i (for simplicity, we take one link between cluster i and j with link cost equal to the average link cost, this only if the link exists), $NL_i > 0$, $Cluster$ be a set of clusters in the network $Cluster = \{1, 2, ..., NC\}$, i and j must be in $Cluster$ set, but $i \neq j$. Then the connectivity per cluster is given by

$$TCCluster_i = \frac{\sum\limits_{j \in Cluster} \overline{C.Cluster_{ij}}}{NL_i}, \quad NL_i > 0, \tag{4.5}$$

where,

$$\overline{C.Cluster_{ij}} = \frac{C.Cluster_i + C.Cluster_j}{2} \tag{4.6}$$

Total Connectivity is the connectivity in the whole network, where we consider all the reachable clusters in it. A reachable cluster is a cluster that can be attainable by other through availabilities border links. Let $TotalC.$ be Total Connectivity representation, and is given by

$$TotalC. = \frac{\sum\limits_{i=1}^{NC} TCCluster_i}{NC}, \tag{4.7}$$

where $TCCluster$ is the Total Connectivity per cluster and NC is the total number of clusters.

Total Connectivity with respect to the link border probability is the connectivity in the whole network, including the probability of two border nodes (link border probability), i, j, corresponding to cluster i and cluster j, mentioned before. This parameter is called $TotalC._{BL}$ and given by

$$TotalC._{BL} = \frac{\sum\limits_{i,j \in Cluster} (\overline{C.Cluster_{ij}}) Pr_{ij}(BL)}{NTL}, \quad NTL > 0, \tag{4.8}$$

where $Pr_{ij}(BL)$ is the average of link border probabilities between cluster i and j, is calculated by Equation (4.2), and NTL is the total number of available connections between clusters, (for simplicity, we take one link between cluster i and j with link cost equal to the average link cost, this only if link exists). A connection between clusters exist if there is at least one link with positive cost between the clusters.

Next step is to analyze the quality parameter similarly to the analysis of the connectivity parameter. In Figure 4.4 we have the different types of quality to analyze in this work.

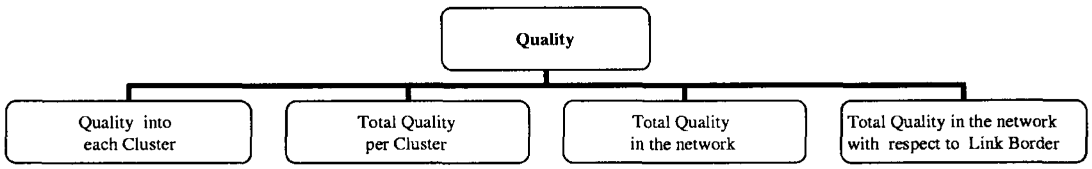


Figure 4.4: Types of Quality at the network generated

Quality into each cluster is the quality related with the amount of available links into the each cluster. Available link is when a link between two nodes has a probability greater than or equal to threshold α. Let $Q.Cluster$ be Quality into each cluster, $Pr_i(link)$ be the link i probability and NAN be the number of reachable nodes in the cluster, except clusterhead node, $Cluster$ be a set of cluster in the network $Cluster = \{1, 2, ..., NC\}$, and i be the cluster analyzed.

$$Q.Cluster = \frac{\sum_{i \in Cluster} Pr_i(link)}{NAN}, \tag{4.9}$$

Quality per cluster is the Quality per cluster respect to the whole network. We consider two clusters to be adjacent whenever they have at least a link with positive cost between them.

Let $TQCluster$ be Total Quality per cluster, $\overline{Q.Cluster}$ be the average of $Q.Cluster$ between two clusters, cluster i and cluster j, where quality of cluster i is analyzing; these clusters must be have an availability link between them, NL_i be the total number of clusters adjacent to cluster i (for simplicity, we take one link between cluster i and j with link cost equal to the average link cost, this only if the link exists), $NL_i > 0$, $Cluster$ be a set of clusters in the network $Cluster = \{1, 2, ..., NC\}$, i and j must be in $Cluster$ set, but $i \neq j$. Then the quality per cluster is given by

$$TQCluster_i = \frac{\sum_{j \in Cluster} \overline{Q.Cluster_{ij}}}{NL_i}, \quad NL_i > 0, \tag{4.10}$$

where,

$$\overline{Q.Cluster_{ij}} = \frac{Q.Cluster_i + Q.Cluster_j}{2} \tag{4.11}$$

Total Quality is the quality in the whole network, where we considering all the reachable clusters in it. A reachable cluster is a cluster that can be attainable by other through availabilities border links. Let $TotalQ.$ be Total Quality representation, and is given by

$$TotalQ. = \frac{\sum_{i=1}^{NC} TQCluster_i}{NC}, \tag{4.12}$$

where $TQCluster_i$ is the Total Quality per cluster and NC is the total number of clusters.

Total Quality with respect to the link border probability is the quality in the whole network including the probability of two border nodes (link border probability), i, j, corresponding to cluster i and cluster j, mentioned before. This parameter is called $TotalQ_{BL}$ and given by

$$TotalQ_{BL} = \frac{\sum_{i,j \in Cluster} (\overline{Q.Cluster_{ij}}) Pr_{ij}(BL)}{NTL}, \quad NTL > 0, \tag{4.13}$$

where $Pr_{ij}(BL)$ is the average of link border probabilities between cluster i and j, is calculate by Equation (4.2), and NTL is the total number of available connections between clusters, (for simplicity, we take one link between cluster i and j with link cost equal to the average link cost, this only if link exists). A connection between clusters exist if there is at least one link with positive cost between the clusters.

Chapter 5

Numerical Results

This chapter presents a proposed algorithm to determine Quality and connectivity in the Ad-Hoc Network; according to the model introduced in Chapter 4.

5.1 Network Generation

In function to the analysis developed previously; we divide this section in cluster generation and link generation.

5.1.1 Cluster Generation

The first step is to generate the nodes in the network. To make this part, we are using a Poisson Distribution, the process to generate the nodes was explained in Chapter 4. Now, we define a square area to ease the network analysis and the number of the clusters, as we describe in Chapter 4. Figure 5.1 shows this. According to this figure, we have here two examples where the first network has four clusters and the second one has nine clusters, note that these numbers have exact square root. Next step is to determine clusterheads and border nodes per cluster, through the Euclidian distance (d_{ij}), by Equation (4.2), as we described in Chapter 4.

5.1.2 Link Generation

Next step is to find the link probabilities between the nodes using Equation (4.3), according to the threshold value, α, t explained in Chapter 3, to the analysis to the network. In this case we have a variable value for α to observed the different behavior of the network.

All nodes must be connected to their respective clusterhead, but if there are nodes not connected to the clusterhead and they have links available to other nodes (these connected to their respective clusterhead), they can be connected by two hops to the clusterhead; All

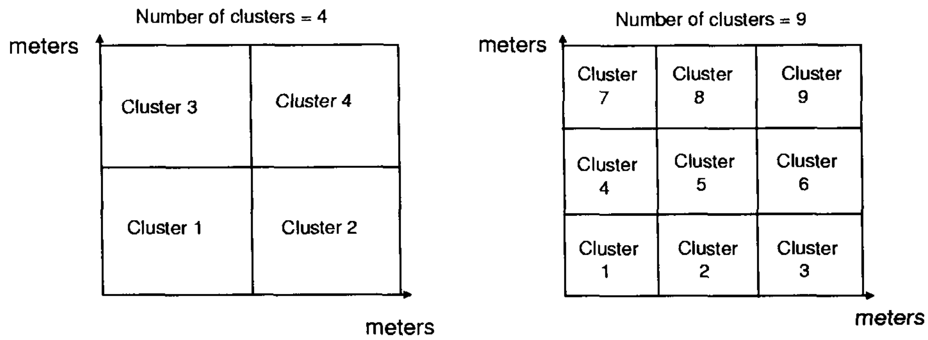


Figure 5.1: Organization area in different numbers of cluster

the clusters must be connected, by border nodes, according to the probability value and α factor. See Figure 5.2

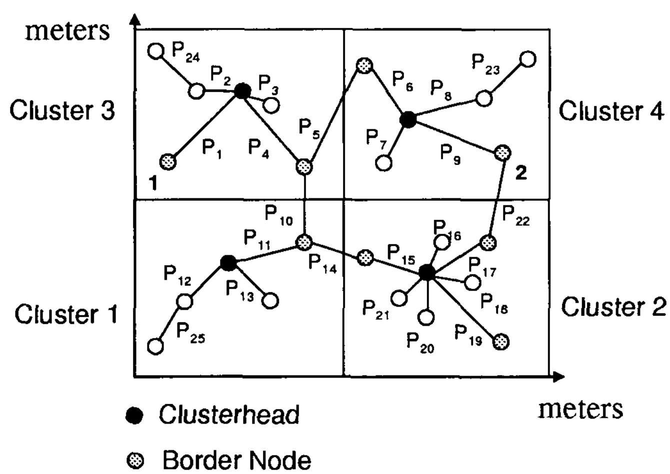


Figure 5.2: Network Connected with 4 Clusters

Now we already have the network, next step is finding connectivity and Quality parameters described in Chapter 4. In the next section, we propose different scenarios to apply the algorithm described before.

5.2 Proposed Scenarios

In this section we propose various types of scenarios where we are apply the concepts described before. The scenarios are

- **Scenario 1**, in this scenario, we compare the cluster performance measures of Connectivity and Quality, in different situations as when a node is added, a link fails and a link is added.
- **Scenario 2**, the objective of this scenario is to compare the connectivity and quality behavior when the total number of nodes and the number of clusters (NC) in the network change.
- **Scenario 3**, in this scenario, we compare the cluster performance measures, Connectivity and Quality, when we vary the velocity of the nodes.
- **Scenario 4**, the objective of this scenario is to compare the connectivity and quality behavior when the threshold α changes.

5.2.1 Scenario 1

The objective of this scenario is to compare the cluster performance measures proposed in this thesis, Connectivity and Quality, in different situations as when a node is added, a link fails and a link is added. These situations are caused for movement of the nodes or users in different directions with different velocities.

We take the network in Figure 5.3, where we apply the algorithm proposed with the following parameters, Total Number of nodes in the network (NTN) = 9, NC=3, $\alpha = 0.5$.

To find Connectivity into each cluster we need to apply Equation (4.4), for example, taking Figure 5.3 for Cluster 1, we have Reachable nodes= 4, remembered that reachable node is the node attainable to another one, Total Nodes into the cluster = 4, therefore Connectivity into the cluster = 1, in percentage we have Connectivity into the cluster (%) = 100%. To find Quality into the cluster we apply Equation (4.9), in Cluster 1 we have a link between 4 to 5, 4 to 8 and 4 to 10 but we have two paths to get nodes 8 and 10 from 4, in this case we need to apply the Trellis Algorithm to find the appropriate path, therefore, for link from 4 to 8 we have $c(p)_1 = 0.72$ and $c(p)_2 = (0.75)(0.9) = 0.648$, we can see that the best path to take is $c(p)_1$, therefore Quality into the cluster is equal to $(0.72 + 0.87 + 0.75)/3 = 0.78$ in percentage we have, Quality into the cluster (%) = 78%. Other parameters to analyzed are Connectivity and Quality per cluster in this case we have link from clusters 1 to 2, 1 to 3, 2 to 1, 2 to 3, 3 to 1 and 3 to 2 and we already found the values of connectivity and Quality intracluster, we can apply Equations (4.5), (4.10)

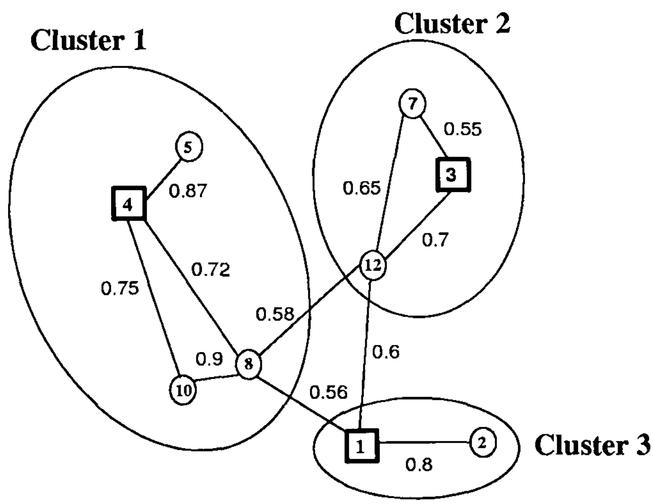


Figure 5.3: Initial Network

and obtain these values. Table 5.1 shows the values for connectivity and Quality into the cluster.

Table 5.1: Connectivity and Quality to Scenario 1, percentage

Cluster	C.Cluster	Q.Cluster	TCCluster	TQCluster
1	100	78	100	74
2	100	63	100	71.5
3	100	80	100	75.2

In Table 5.1 we also have the values of Total Connectivity and Quality obtained to equations (4.7) and (4.7). We have Total Connectivity equal to 100% ,Total Quality equal to 73.4%, Total connectivity with respect border link (BL) equal to 58% and Total Quality with respect border link (BL)equal to 42.6% these values are acceptable to performance the networks because the clusters are totally attainable with acceptable value of quality. But what happen if a node or link is added to the network? or if a Link failure? Tables 5.2, 5.3, 5.4 and 5.5 show the change of parameters, according to Figure 5.4.

In conclusion, the algorithm proposed in this work provides good stability for connectivity and quality due to the minimum changes for these measures in the four cases studied in this section, hence, we can say that this algorithm is better than ABCP and DCMA because the change in the cluster configuration, due to the movement of the nodes, is too low to compare with the other two, this factor causes stability in the cluster performance

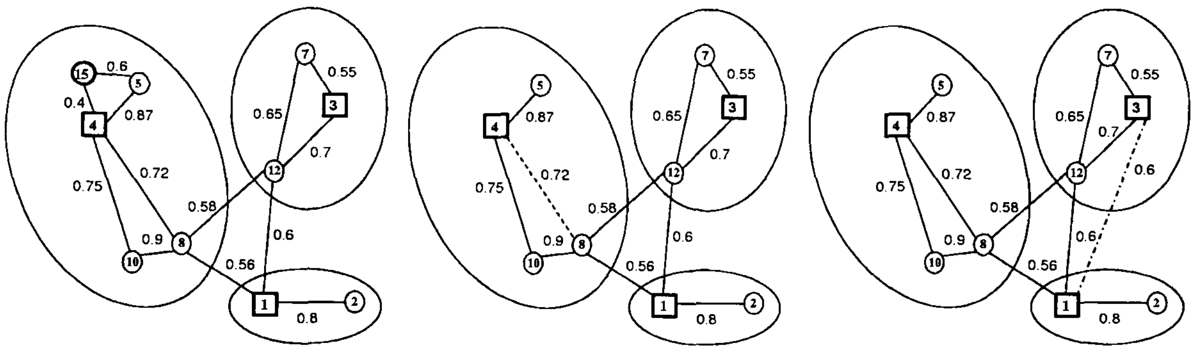


Figure 5.4: Situations Scenario 1

Table 5.2: Connectivity and Quality if a Node is added to the Network, percentage

Cluster	C.Cluster	Q.Cluster	TCCluster	TQCluster
1	100	71.55	100	73
2	100	63	100	69.28
3	100	80	100	75.2

Table 5.3: Connectivity and Quality if a Link Fails, percentage

Cluster	C.Cluster	Q.Cluster	TCCluster	TQCluster
1	75	73.5	87.5	72.37
2	100	63	93.7	69.75
3	100	80	93.7	74.125

Table 5.4: Connectivity and Quality if a Link is added, percentage

Cluster	C.Cluster	Q.Cluster	TCCluster	TQCluster
1	100	78	100	74
2	100	63	100	71.5
3	100	80	100	75.2

Table 5.5: Total Cluster Performance, percentage

Parameter	Initial Network	Node is Added	Link Fails	Link is Added
$TotalC.$	100	100	100	100
$TotalQ.$	73.4	72.493	72.493	73.4
$TotalC_{BL}$	58	58	53	58.813
$TotalQ_{BL}$	42.6	42.05	41.7	43.2425

measure, and is based on DDCA clustering algorithm.

5.2.2 Scenario 2

An Ad-Hoc network has some important parameters that are related to the behavior of the performance measures. These parameters are NTN, which corresponds to the total number of users in the network, NC, the number of clusters taking to analyze the network, α, the threshold to probabilities of the links and the velocity v, the motion of the nodes per unit of time, in this case per simulation time interval (ITS). Realization Number is the amount of ITS; ITS can be calculated in seconds as, for 2m/ITS,

$$\overline{v} = 1m/ITS, \tag{5.1}$$

Note that we take the average velocity in Equation (5.1) because users move with different velocities.

In this scenario, we vary NTN and NC to compare the behavior of Connectivity and Quality in the network. Table 5.6 shows the parameters of Scenario 2.

Table 5.6: Parameters to Scenario 2

Parameter	Value
Number of nodes (NTN)	100, 200, 400
Number of Clusters (NC)	4, 9, 16
α	0.3
Velocity	0 - 2 m/ITS

Figures 5.5, 5.6 and 5.7 show the cluster connectivity and quality into the cluster, performance into the Cluster, behavior obtained with Equations (4.4) and (4.9) through

the ten (10) realization for 100, 200, and 400 nodes to 4, 9, 16 clusters.

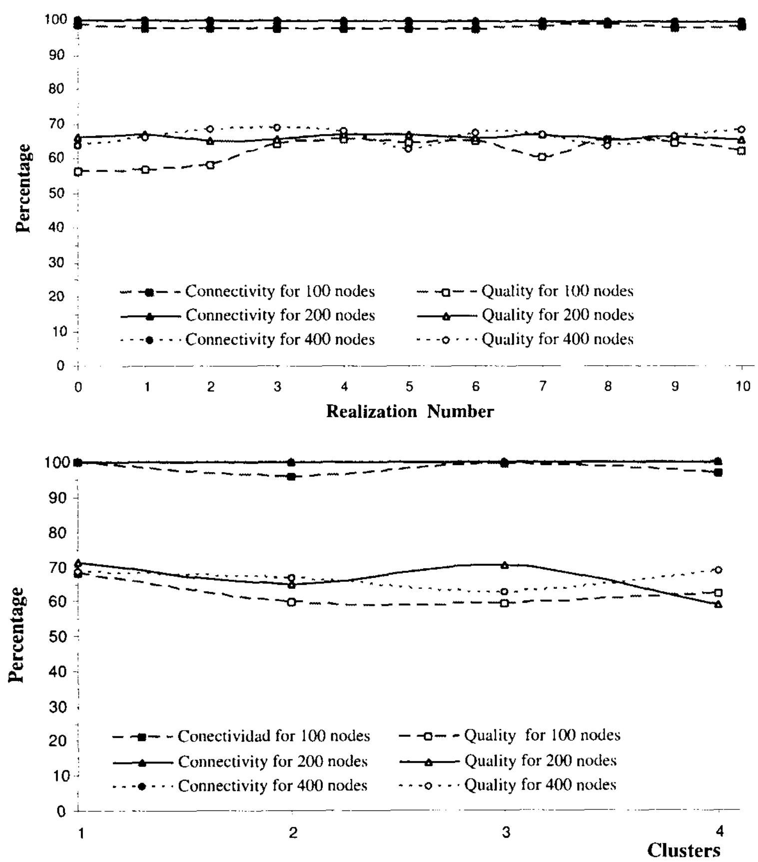


Figure 5.5: Performance into the Cluster(4 Clusters)

In each figure we have two kinds of graphics, one is with respect to Realization Number and the other one is with respect to the quantity of clusters. Compare the behavior of these Figures 5.5, 5.6 and 5.7, we have that the connectivity and quality is unstable when we have 16 clusters and is stable and increases when we have 4 or 9 cluster, therefore the network is better if operate with these number of cluster. Note that connectivity and quality have zero values when the network has 100 nodes and $NC = 16$, in Figure 5.7, this case is presented when the cluster does not have reachable nodes into the cluster or when the cluster is empty, not have any node.

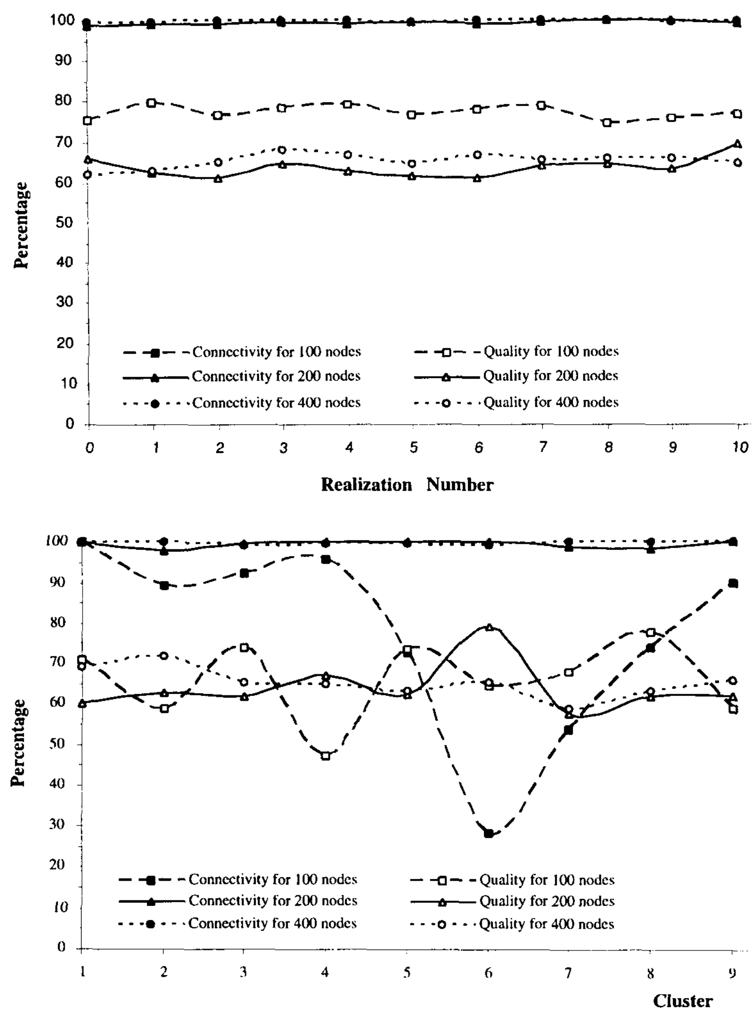


Figure 5.6: Performance into the Cluster (9 Clusters)

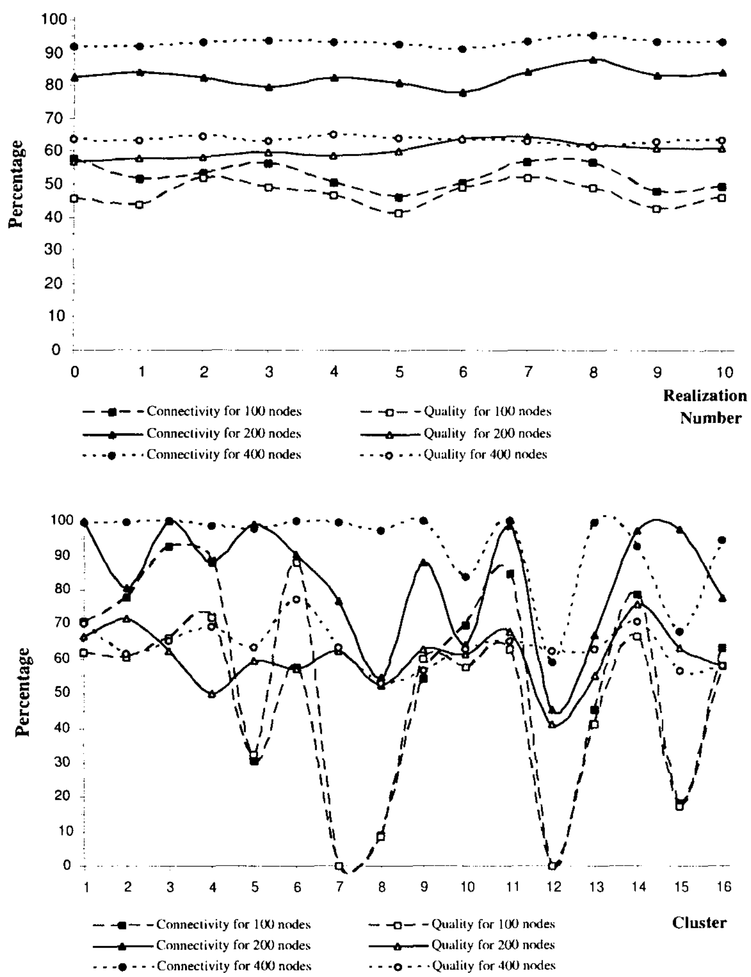


Figure 5.7: Performance into the Cluster (16 Clusters)

Figure 5.8, 5.9, 5.10 show the Total Connectivity and Quality in the network obtained with Equations (4.7) and (4.12) respectively, in other words, Total Cluster Performance.

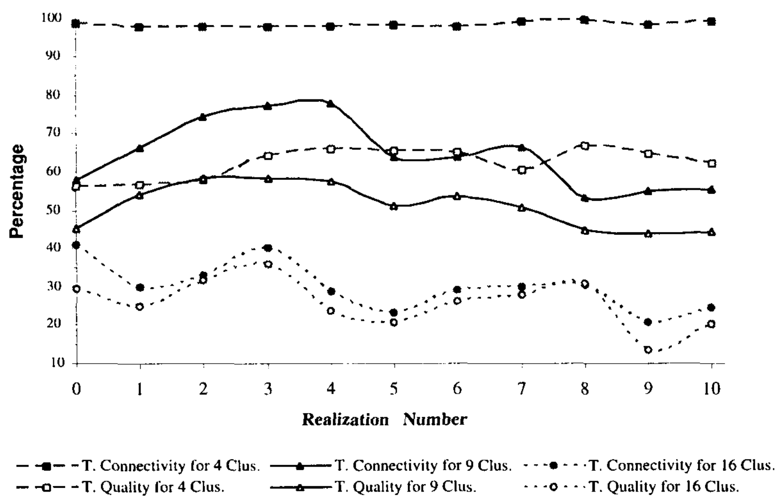


Figure 5.8: Total Cluster Performance for 100 nodes

In conclusion, a network with 200 or 400 nodes and 4 or 9 cluster is stable, because it has reasonable values to make good performance and service to users, on Connectivity and Quality.

In Figures 5.11, 5.12, 5.13 we have the total connectivity and Quality including the value of probability to border links, obtained with Equations (4.8) and (4.13), these parameters depends the value of α, therefore if α is increasing Quality is increasing too, but the connectivity in the network decrease in order to the value of link' probability. In section later we explain better this concept.

In Figure 5.11, we compare Connectivity and Quality respect to the number of clusters for 100 nodes. In Connectivity we can see that is better than the others, when we have 4 clusters, while in quantity is better when we have 16 clusters. This difference is present due to value of link probabilities and the instability of the network when it have 100 nodes.

Figures 5.12 and 5.13 show the Connectivity and Quality respect to the number of clusters for 200 and 400 nodes, respectively. When $NTN = 200$ and $NTN = 400$, the Connectivity and Quality are maintained stable with 4, 9, and 16 clusters, being better when $NC = 4$ and $NC = 9$.

In order to the results obtained in this section, we are using 9 clusters, and 200 nodes to others simulations, and we are varying the velocity and α parameters.

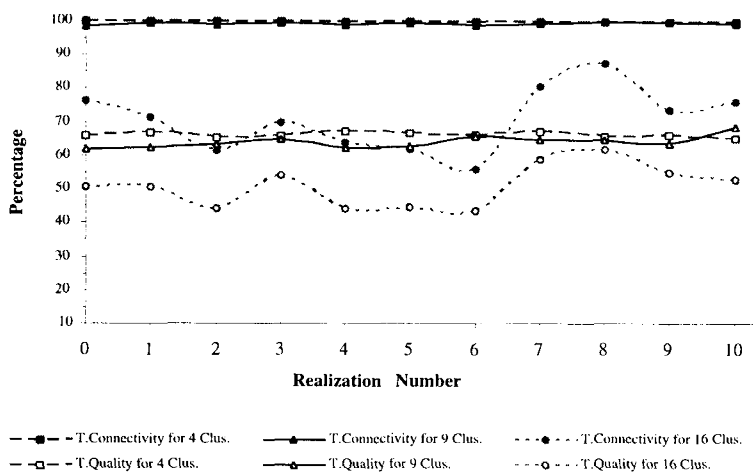


Figure 5.9: Total Cluster Performance for 200 nodes

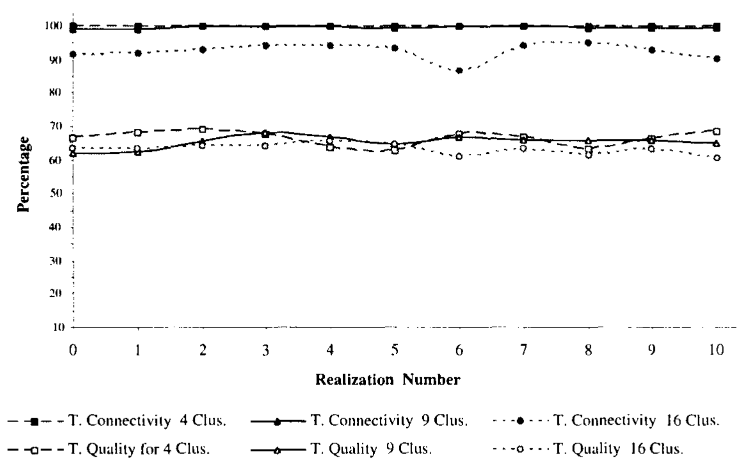


Figure 5.10: Total Cluster Performance for 400 nodes

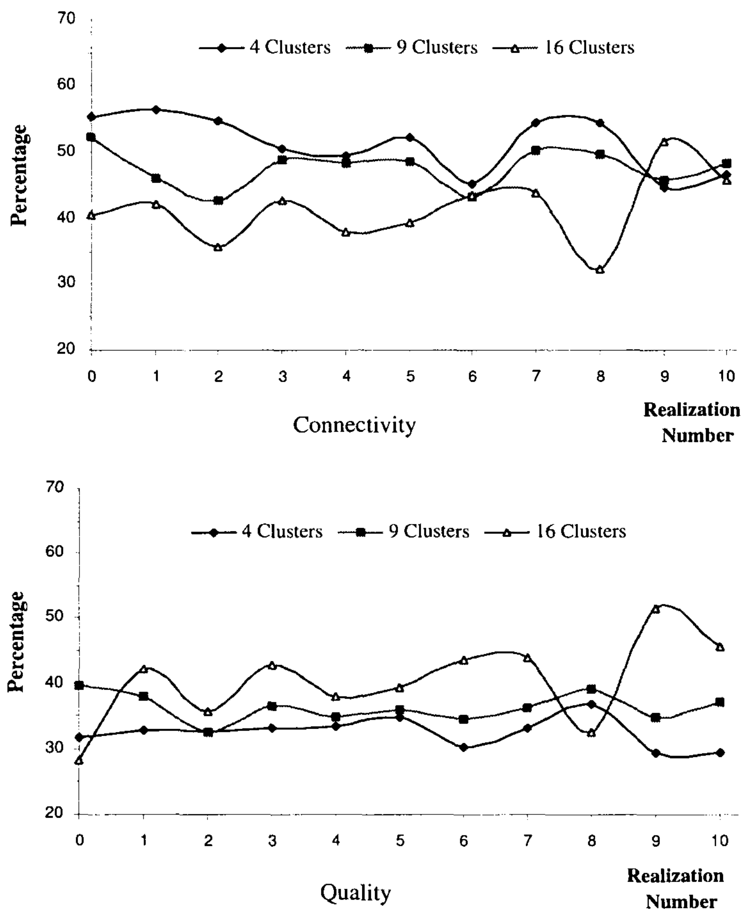


Figure 5.11: $TotalC_{BL}$ and $TotalQ_{BL}$ for 100 nodes

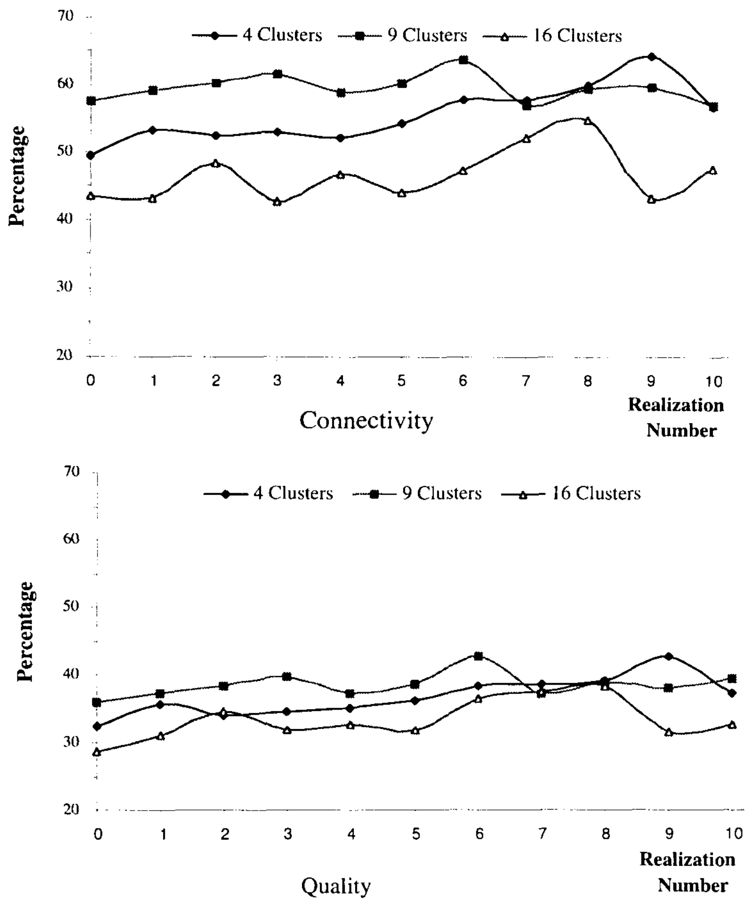
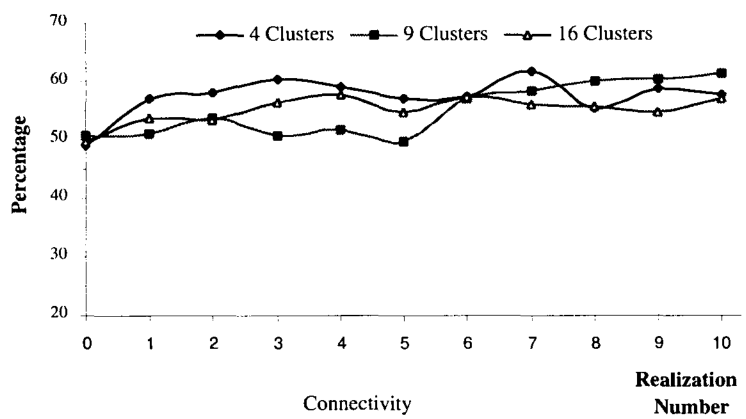


Figure 5.12: $TotalC_{BL}$ and $TotalQ_{BL}$ for 200 nodes



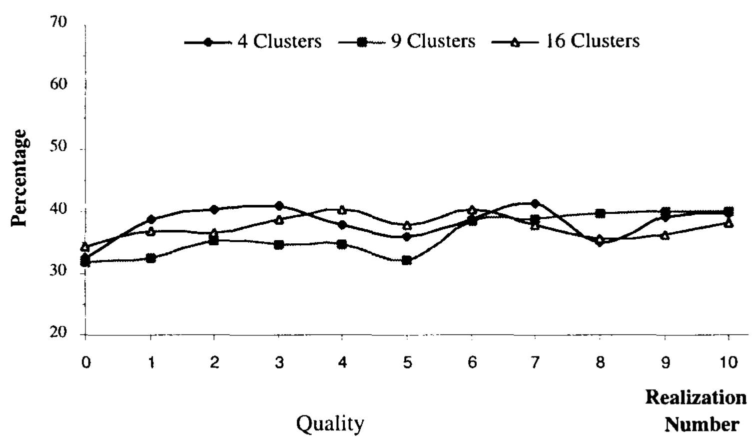


Figure 5.13: $TotalC_{BL}$ and $TotalQ_{BL}$ for 400 nodes

5.2.3 Scenario 3

In this section we vary the velocity of the nodes and look the network behavior, Quality and Connectivity parameters. Table 5.7 shows the parameters to need in this scenario.

Table 5.7: Parameters to Scenario 3

Parameter	Value
Number of nodes	200
Number of Clusters (NC)	9
α	0.3
Velocity	0 - 2 m/ITS, 0 - 5m/ITS, 0 - 10m/ITS

Table 5.8 shows the values of Total Connectivity and Quality in the network and they are associate to Figure 5.14 for velocity between 0 - 2m/ITS, 0 - 5m/ITS, 0 - 10m/ITS and obtained with Equations (4.7) and (4.12) respectively .

Table 5.8: Total Connectivity and Quality, in percentage

Realization Number	0-2 m/ITS		$0-5m/ITS$		$0-10m/ITS$	
	Connectivity	Quality	Connectivity	Quality	Connectivity	Quality
0	98.64742	61.81192	98.647422	61.81192	98.64742	61.81192
1	99.42128	62.2073	99.02817	62.42941	98.77736	65.23647
2	99.2647	63.29277	99.50617	65.8485	86.97916	56.1154
3	99.62606	64.74921	100	65.57725	98.42662	65.43232
4	99.09	62.40776	100	65.31267	98.07692	61.92681
5	99.51923	63.16044	99.38271	68.94980	99.14461	62.79284
6	99.2647	66.03222	87.90706	60.11681	98.91304	61.18587
7	99.54732	64.81965	98.60890	69.41936	98.25182	64.17968
8	100	65.03438	96.90752	66.01729	99.30554	60.87383
9	100	63.88987	97.63374	68.15112	97.92663	58.07583
10	99.30555	68.9732	99.34640	67.19004	100	62.6698

In conclusion, total connectivity and Quality do not change significantly if the velocity of the nodes change between whatever value, so we can consider the network stable in this condition. We can verify this condition too, in Connectivity and Quality per cluster to each velocity described before. Table 5.9, and Figure 5.15 show the simulation results for connectivity and quality per cluster for velocity between 0 to 2 m/ITS, 5 m/ITS and 10 m/ITS, obtained with Equations (4.5) and (4.10).

Table 5.9: Connectivity and Quality per Cluster, in percentage

Cluster	$0 - 2m/ITS$		$0 - 5m/ITS$		$0 - 10m/ITS$	
	Connectivity	Quality	Connectivity	Quality	Connectivity	Quality
1	99.82841	62.69444	99.61017	66.34834	99.3007	64.94615
2	98.86287	61.9943	99.19712	66.07436	99.84050	63.70283
3	99.25202	66.29117	99.77272	67.40384	98.84763	62.66667
4	99.80429	63.442418	98.17071	64.93488	98.71785455	63.5064
5	99.55349	64.82670	98.10987	66.15687	98.66490	62.55338
6	99.92025	70.44012	99.88636	69.71845	98.73670	62.44238
7	99.01180	61.00960	88.18180	56.75858	89.41648	55.83087
8	99.00214	61.17417	98.36065	64.701518	97.47360	60.2472
9	99.59892	66.07330	99.866309	67.668428	98.09635	60.71385

Another parameter to consider is the connectivity and Quality into the each cluster, described in Chapter 4, Tables 5.10 and 5.11 and Figure 5.16, respectively, show these parameters and obtained with Equations (4.4) and (4.9). So, we have two kinds of graphics here, one is connectivity and Quality with respect to quantity of clusters, where can see the internal behavior of each cluster in the net in the time, here we take the average value to each cluster through the ITS (t_0 to t_{10}); the other graphic is connectivity and quality with respect the time, in this figure we want to see the behavior for all the clusters in each instant of time, in other words is the group behavior to level the connectivity and quality intracluster. To find this behavior, we get the average value to parameter (connectivity and quality)in each time, taking all the clusters, not each cluster.

The last parameter to consider is the total Connectivity and Quality with respect the border links. We are getting the same same analysis than before. Figure 5.17 and Table 5.12 show the results to these parameters obtained with Equations (4.8) and (4.13).

If we compare these parameters, we can see that the change in the network is minimum, and in this case the real parameter need to change α. Next section we are to analyze this threshold.

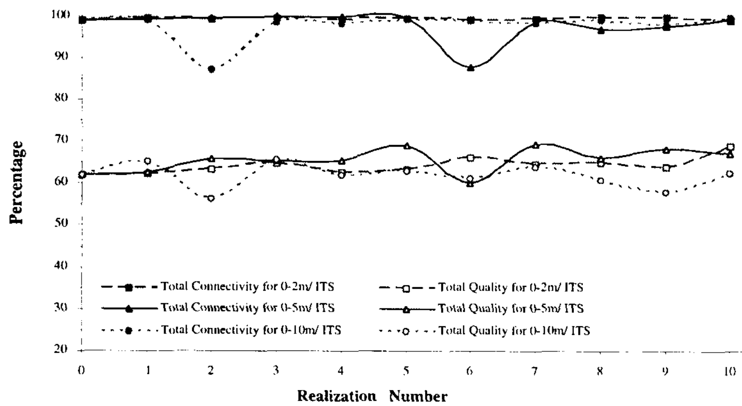


Figure 5.14: Total Connectivity and Quality in the network

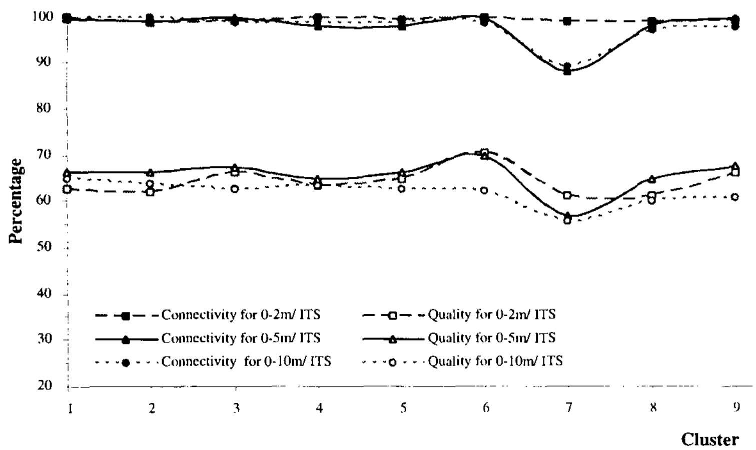


Figure 5.15: Connectivity and Quality per Cluster

Table 5.10: Connectivity and Quality into the Cluster with respect to NC, in percentage

Cluster	$0-2m/ITS$		$0-5m/ITS$		$0-10m/ITS$	
	Connectivity	Quality	Connectivity	Quality	Connectivity	Quality
1	100	60.10813	99.22035	66.57525	100	65.42427
2	97.96499	62.62206	100	66.33104	100	64.40204
3	99.52153	61.77737	99.54545	64.82858	98.33576	61.54854
4	100	66.96730	98.60139	63.27384	98.60139	63.27384
5	100	62.19057	96.43181	65.41488	98.95334	62.77172
6	100	78.98789	100	73.62718	98.71900	63.04675
7	98.82575	57.48388	91.56045	58.64609	96.94349	61.33388
8	98.39572	61.88713	99.46524	64.95231	96.57658	58.98310
9	100	61.70911	100	66.04710	98.13169	60.74140

Table 5.11: Performance into the Cluster respect the Realization Number, in percentage

Realization Number	$0-2m/ITS$		$0-5m/ITS$		$0-10m/ITS$	
	Connectivity	Quality	Connectivity	Quality	Connectivity	Quality
0	98.60566	61.18272	98.60566	61.18272	98.60566	61.18272
1	99.30555	61.58483	98.92241	62.01882	98.76160	65.07166
2	99.346405	62.90498	99.25925	65.01091	97.17592	62.93984
3	99.57264	64.62250	100	65.44078	98.03675	64.87231
4	98.98785	61.20150	100	65.50877	98.29059	61.83059
5	99.57264	62.70645	99.25925	68.96746	98.91534	63.37070
6	99.34640	65.76217	94.92174	65.58544	98.55072	60.51577
7	99.58847	64.28597	98.50970	69.49766	98.33054	64.06511
8	100	64.50980	96.42338	66.02626	99.30555	61.52573
9	100	63.23108	97.53086	67.68251	97.23553	58.29758
10	99.20634	69.23774	99.34640	67.20233	100	62.63697

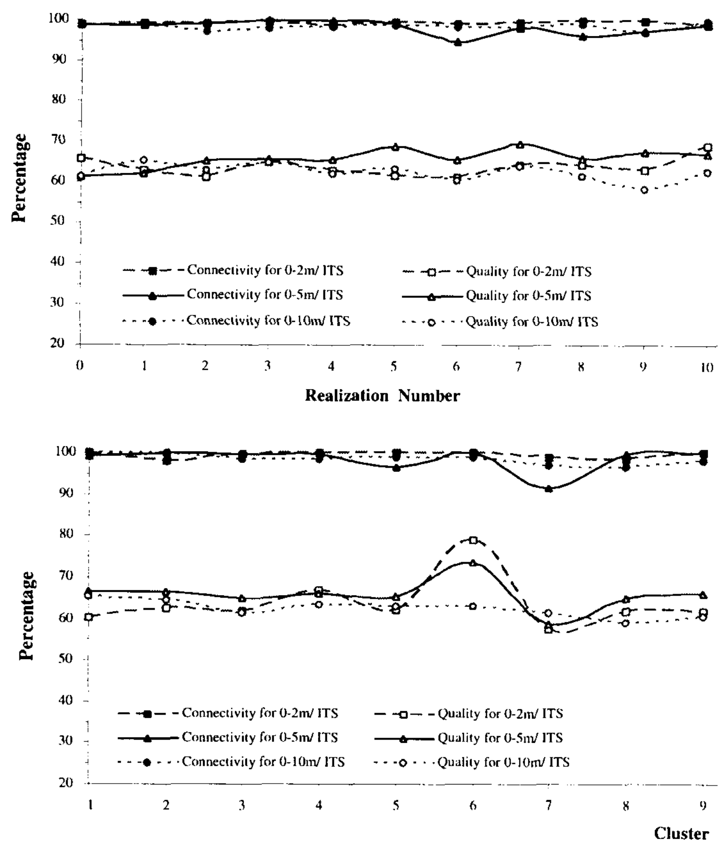


Figure 5.16: Connectivity and Quality into the Cluster.

Table 5.12: Cluster Performance respect to the Link Border Probabilities, in percentage

Realization Number	$0-2m/ITS$		$0-5m/ITS$		$0-10m/ITS$	
	Connectivity	Quality	Connectivity	Quality	Connectivity	Quality
0	57.44292	36.03996	57.44269	36.0396	57.03658	37.36589
1	59.22187	37.24387	63.70505	40.54836	56.94727	37.45098
2	60.23818	38.35391	57.82966	38.37471	58.09983	37.51121
3	61.45473	39.68302	59.82855	39.23254	54.04888	35.83855
4	58.73382	37.30003	62.083	40.59906	50.34132	31.70408
5	60.32093	38.53998	56.80797	39.36303	56.66716	35.82065
6	63.70971	42.60845	53.36688	36.60552	54.31033	33.86579
7	57.11020	37.27044	54.21196	38.16862	56.20868	36.85094
8	59.37675	38.85315	50.32539	34.62866	57.50412	35.59881
9	59.74374	38.16025	52.26161	36.58152	51.95545	30.75804
10	56.92325	39.55626	54.52766	36.68457	52.54386	32.85386

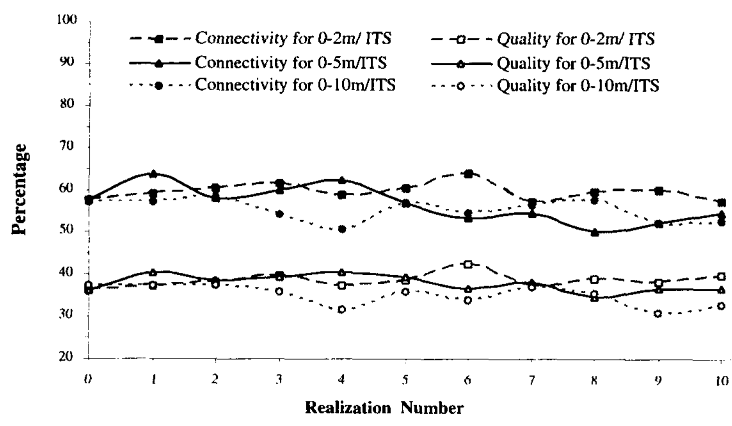


Figure 5.17: Total Connectivity and Quality respect Link Borders

5.2.4 Scenario 4

Now, we varying the bound α, described in chapter 3, and looking the behavior network through performance parameters as Quality and Connectivity.

Table 5.13: Parameters to Scenario 4

Parameter	Value
Number of nodes	200
Number of Clusters (NC)	9
α or Threshold	0.2, 0.3, 0.6
Velocity	0 - 2 m/ITS

Table 5.14 shows the values of Total Connectivity and Quality in the network and they are associate to Figure 5.18 for α equal to 0.2, 0.3, 0.5.

Table 5.14: Total Connectivity and Quality, in percentage

Realization Number	$\alpha = 0.2$		$\alpha = 0.3$		$\alpha = 0.6$	
	Connectivity	Quality	Connectivity	Quality	Connectivity	Quality
0	100	61.83304	99.59892	66.07330	45.55241	46.12800
1	100	62.827792	99.42128	62.2073	48.86585	47.96380
2	100	63.116736	99.2647	63.29277	52.21695	47.92477
3	100	62.25415	99.62606	64.74921	66.46234	64.16903
4	100	63.046763	99.09	62.40776	69.70929	61.19918
5	100	65.64596	99.51923	63.16044	71.66161	63.104076
6	100	63.046763	99.2647	66.03222	80.99145	72.82364
7	100	64.22478	99.54732	64.81965	75.19332	64.13812
8	100	65.04251	100	65.03438	85.83769	71.42515
9	100	64.31372	100	63.88987	82.52168	71.26774
10	99.30555	68.45387	99.30555	68.9732	76.93762	66.67020

In conclusion, total connectivity and Quality do not change significantly if the $\alpha = 0.2$ or 0.3 therefore we can consider the network stable in this condition, but if $\alpha = 0.6$ the connectivity parameter decreases in order to value of link probabilities, where if α is increasing the links between nodes probably do not exist because only must be exist the link that probability value is greater than the bound, while quantity has a minimum increment in its value.

Other parameters need to be analyzed are in Connectivity and Quality per cluster to each value of α, 0.2, 0.3, 0.6. Table 6.1, and Figure 5.19 show the simulation results.

Table 5.15: Connectivity and Quality per Cluster, in percentage

Cluster	$\alpha = 0.2$		$\alpha = 0.3$		$\alpha = 0.6$	
	Connectivity	Quality	Connectivity	Quality	Connectivity	Quality
1	100	63.14710	98.64742	61.81192	27.98708	25.65743
2	99.67532	61.60495	99.82841	62.69444	81.02355	70.11530
3	99.83766	66.33184	98.86287	61.9943	85.23844	72.39386
4	100	63.35212	99.25202	66.29117	57.51671	51.31562
5	99.91883	63.85617	99.80429	63.44241	75.14470	70.93458
6	100	70.58761	99.55349	64.82670	89.77118	74.72836
7	100	61.10270	99.92025	70.44012	50.89241	44.37242
8	100	60.68256	99.01180	61.00967	73.65330	71.64349
9	100	66.45420	99.00214	61.17417	77.27736	72.59560

Similarly to Total Connectivity and Quality, the behavior in Connectivity and Quality per cluster is very stable in $\alpha = 0.2$ and 0.3, but when $\alpha = 0.6$ the Connectivity and Quality per cluster decrease and is to variable due to link probabilities explained before.

Other parameter to consider is the connectivity and Quality into the each cluster, described in Chapter 4, Tables 5.16 and 5.17 and Figure 5.20, respectively, show these parameters. So, we have two kinds of graphics here, one is connectivity and Quality with respect to quantity of clusters, where can see the internal behavior of each cluster in the net in the time, here we take the average value to each cluster through the time (t_0 to t_{10}); the other graphic is connectivity and quality with respect the time, in this figure we want to see the behavior for all the clusters in each ITS, in other words is the group behavior to level the connectivity and quality intracluster. To find this behavior, we get the average value to parameter (connectivity and quality)in each time, taking all the clusters, not each cluster.

Connectivity and Quality into the cluster that the parameters into the cluster that the network is to stable and values of connectivity and quality are acceptable when α is equal to 0.2 and 0.3, but when α is equal to 0.6 the value of connectivity decrease, in other words not all of nodes are reachable nodes, therefore some of the nodes cannot to communicate to others, but the Quality parameter is increasing because the value of the bound makes that links with low probability (regular communication) do not exist.

The last parameter to consider is the total connectivity with respect the border links. We are getting the same same analysis than before. Figure 5.21 and Table 5.18 show these parameters.

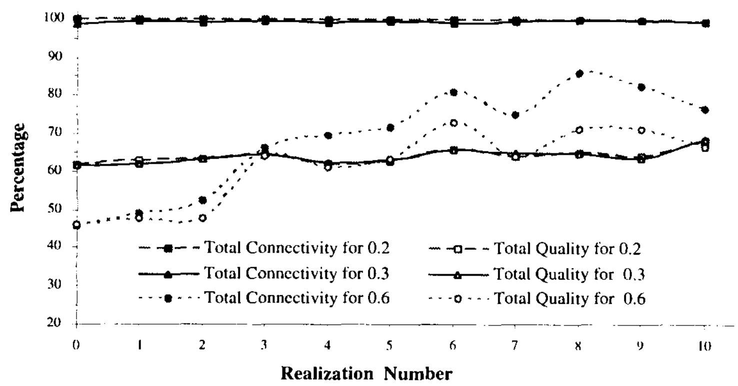


Figure 5.18: Total Connectivity and Quality in the network

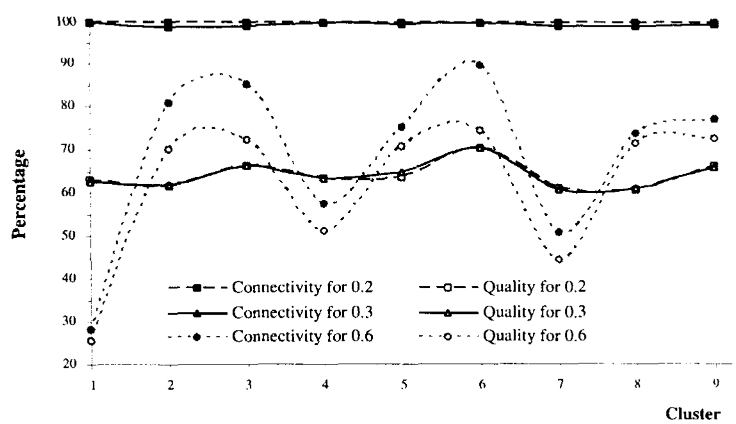


Figure 5.19: Connectivity and Quality per Cluster

Table 5.16: Connectivity and Quality into the Cluster with respect to NC, in percentage

Cluster	$\alpha = 0.2$		$\alpha = 0.3$		$\alpha = 0.6$	
	Connectivity	Quality	Connectivity	Quality	Connectivity	Quality
1	100	60.53538	100	60.10813	69.64137	72.57086
2	99.35064	62.84418	97.96499	62.62206	85.24545	69.49722
3	100	61.25618	99.52153	61.77737	78.32066	70.38609
4	100	67.16525	100	66.96730	83.27818	71.18121
5	100	60.05216	100	62.19057	73.88812	71.28943
6	100	79.97082945	100	78.98789	99.06698	79.30605
7	100	58.02942764	98.82575	57.48388	59.57664	68.09176
8	98.39572	61.23632	98.39572	61.88713	71.74688	71.71618
9	100	62.30482	100	61.70911	76.43388	71.68997

Table 5.17: Performance into the Cluster respect the Realization Number, in percentage

Realization Number	$\alpha = 0.2$		$\alpha = 0.3$		$\alpha = 0.6$	
	Connectivity	Quality	Connectivity	Quality	Connectivity	Quality
0	100	61.18322	98.60566	61.18272	67.66335	69.62051
1	100	62.22853	99.30555	61.58483	69.98640	71.85870
2	100	62.69760	99.34640	62.90498	73.94541	72.045464
3	100	64.53049	99.57264	64.62250	76.58395	73.33765
4	100	60.99386	98.98785	61.20150	73.30406	68.80637
5	100	62.63456	99.57264	62.70645	78.91116	70.28761
6	100	65.20542	99.34640	65.76217	80.19038	72.48775
7	100	63.92574	99.58847	64.28597	79.87368	72.29879
8	100	64.68368	100	64.50980	85.34244	70.96637
9	100	63.80497	100	63.23108	81.80586	71.31899
10	99.20634	68.92748	99.20634	69.23774	84.52442	76.19586

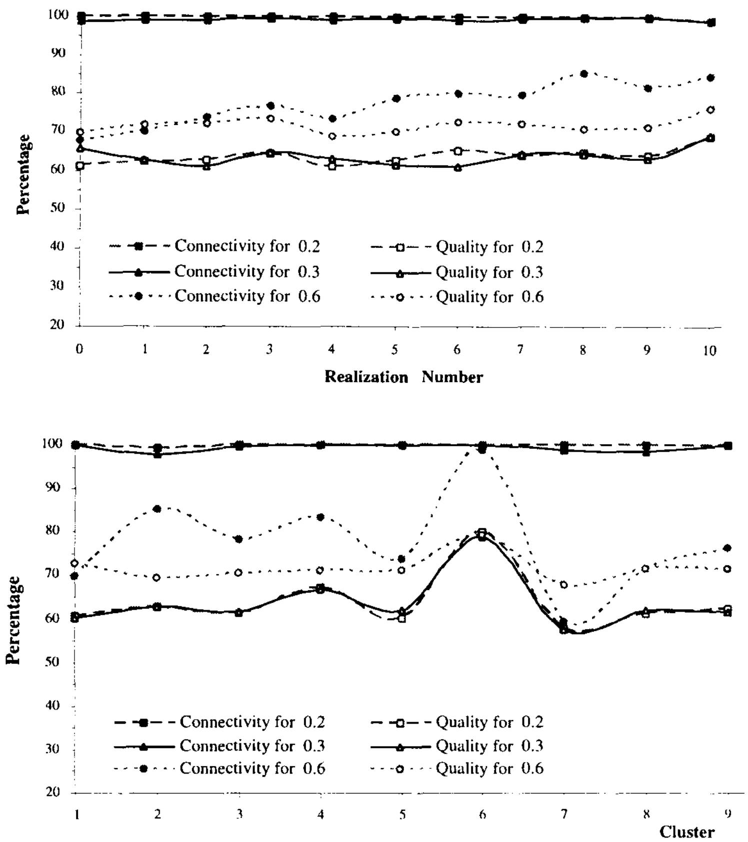


Figure 5.20: Connectivity and Quality into the Cluster.

Table 5.18: Cluster Performance respect to the Link Border Probabilities, in percentage

Realization Number	$\alpha = 0.2$		$\alpha = 0.3$		$\alpha = 0.6$	
	Connectivity	Quality	Connectivity	Quality	Connectivity	Quality
0	55.706	34.441	57.44292	36.03996	53.87	53.78
1	55.80842	35.33898	59.22187	37.24387	58.76053	57.27212
2	56.74489	35.79410	60.23818	38.35391	63.40208	57.62363
3	56.48323	36.32814	61.45473	39.68302	62.06512	58.39261
4	54.57907	34.36860	58.73382	37.30003	64.66401	56.60072
5	55.17588	35.08212	60.32093	38.53998	69.74414	61.88203
6	58.28251	38.32076	63.70971	42.60845	69.30963	62.29020
7	52.23261	33.55477	57.11020	37.27044	66.55868	56.51380
8	54.50324	35.50752	59.37675	38.85315	66.04945	55.50808
9	52.87626	33.97657	59.74374	38.16025	62.99046	54.67180
10	50.93562	35.03633	56.92325	39.55626	68.28461	59.66360

We can appreciate in this last cluster performance measures, that, when α is equal to 0.6 the value of connectivity and quality respect to border links increase according to the value of link probability because these probabilities have major values than the other when α are equal to 0.2 and 0.3.

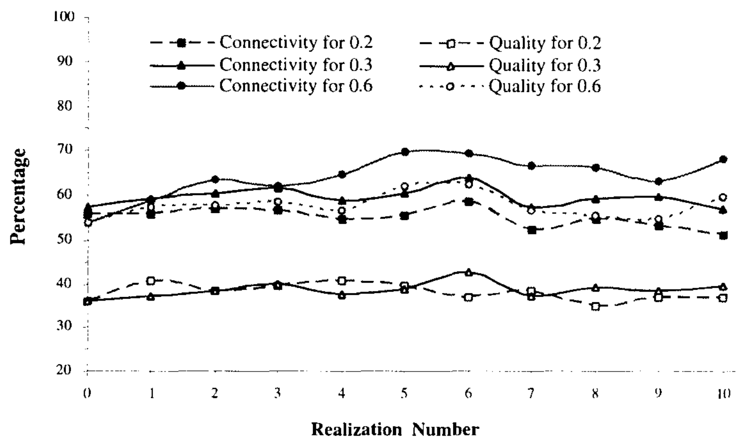


Figure 5.21: Total Connectivity and Quality with respect to Link Borders

Chapter 6

Conclusion and Further Research

This thesis presents in Section 6.1 the conclusions of the thesis, and Section 6.2 some future research.

6.1 Conclusions

Nowadays, the research of Ad-Hoc networks are increase in order to implement new technologies in Telecommunication systems. Main characteristics of these networks are, not having fixed routers and all nodes are capable of movement and have responsibilities for organizing and controlling the network, in other words, networks are self-creating, self-organizing and self-administering. At this moment exist several algorithms that make these responsibilities easily. For these thesis we chose an algorithm based in cluster of nodes, as Distributed Dynamic Clustering Algorithm (DDCA) explained in Chapter 3. Also we used a Trellis Algorithm to find paths between two nodes (origin, destine), using probabilities as links cost.

On Chapter 4 we presented a method to generate the Ad-Hoc Network through DDCA and Trellis Algorithm with different number of nodes, number of clusters, velocities and threshold parameter (α), and evaluate connectivity and quality in the network. The model is based in Trellis algorithm, when has not attainable node by the clusterhead in a hop, but it can be attainable by two hops, and DDCA algorithm to maintain an effective topology in an interval of time, through α parameter, used in the link probabilities as a threshold for them. All of these in order to improve the network efficiency and quality in the performance measure evaluated in this work. Figure 6.1 shows an example of Trellis and DDCA algorithm application. In this case $\alpha = 0.5$, therefore only the links with probabilities equal or higher to α are available. We are to communicate node 5 to node 3; to make this operation, we have several paths with different probabilities, calculated by Equation (3.4), therefore, in the moment when different cases, as new node is added, link failure and new link, are presented, the topology will be stable, and the time response when these kind of cases occur, is minimum compared with other algorithms, (DMAC and ABCP for example) because we have calculated the other paths with minor probabilities.

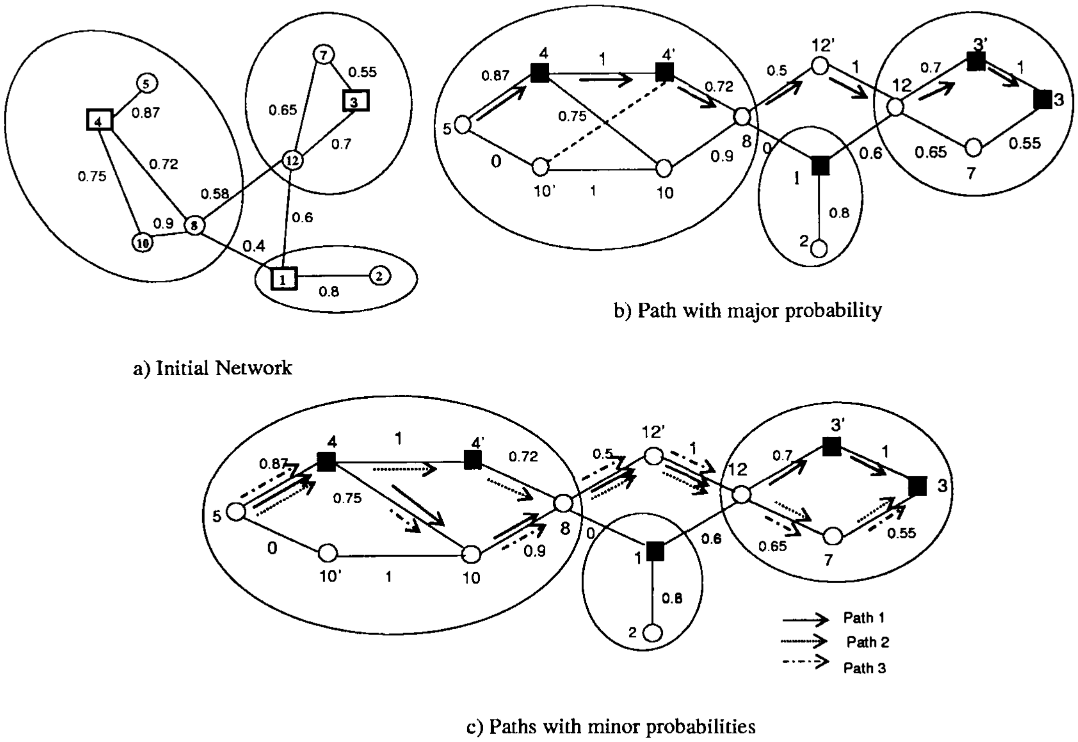


Figure 6.1: Trellis and DDCA Algorithm Application

The result of the thesis is the connectivity and quality evaluating in different forms, into the cluster, per cluster, total, and depends to the link border probability.

According to the results obtained in Chapter 5, and the simulation, Table 6.1 shows the parameters α and Number of cluster (NC) where the network is stable, and connectivity and quality minimum values to get acceptable performance of the network. This table is for the network operator or service provider.

In the system design, it is very important to pay attention to the threshold parameter α because if it is increased, quality increases as well, but connectivity decreases. Node Velocity changes do not cause important variations in the performance measures.

Other important conclusion is that Trellis Algorithm is more difficult to implement than others (as Bellman Ford, Dijkstra), because this algorithm finds all the possible paths between two nodes (origin, destine), while in the others we can find only the path with major cost. Therefore, in case when a link fail in a path chose for communication, for example, it is not necessary to recalculate the paths again because they (origin, destine)

Table 6.1: Acceptable Values to Connectivity and Quality

Parameter	Value
α or Bound	0.3 - 0.5
Number of Clusters (NC)	4 or 9
Connectivity into the cluster	90% min.
Connectivity per cluster	95% min.
Total Connectivity	95% min.
Total Connectivity depend to Link Border	95% min.
Quality into the cluster	60% min.
Quality per cluster	60% min.
Total Quality	60% min.
Total Quality depend to Link Border	60% min.

have other options of the paths calculated before. So, the response time, when these kind of cases occur, is minimum compared with other algorithms where is necessary calculate the path again.

6.2 Further Research

According to the work done in this thesis, the following ideas can be suggested for further research.

- Analysis to change the form and the organization or distribution the clusters in the network generated in this work. For example, taking circular clusters or different forms to the area (not square). Only needs to change some conditions in the Model proposed in this thesis, in the part of network generated, on Chapter 4.
- To make more real the analysis of connectivity and quality, it is convenient the study of the mechanism of signal propagation as Reflection, Diffraction and Scattering, and its relation with links probabilities. In other words, the study of the behavior if a wave impinges upon an object where its dimensions are greater lower than wavelength, or if the obstacle has a surface with sharp irregularities.
- Other parameters to need consider are fading, interferences with other nodes, and its performance when a Social Grouping Behavior is apply, in order to related to the

Connectivity and Quality studied in this thesis and find other important parameters not finding in this research, such transmission and received power, loss in the communication and others.

Bibliography

[1] Ting Chao, "An Access Based Clustering Protocol for Multihop," *IEEE. Journal on Select Areas in Communications,* Vol. 19, No.7, pp. 1201-1210, July 2001.

[2] Satyabrata Chakrabarti, "QoS issues in Wireless Ad Hoc Networks," *IEEE Communications Magazine,* Vol. 39, No. 2. pp. 142-148, February 2001.

[3] Bruce MacDonald, "A Mobility - Based Framework for Adaptive Clustering in Wireless Ad Hoc Networks," *IEEE Journal on Select Areas in Communications,* Vol. 17, No.8, pp. 1466-1487, August 1999.

[4] C-K. Toh, *Ad Hoc Mobile Wireless Networks,* Prentice Hall, 2002.

[5] Charles E. Perkins, *Ad Hoc Networking,* Addison Wesley, 2001.

[6] P. Basu, N. Khan, and T. D. C. Little, "A Mobility Based Metric for Clustering in Mobile Ad Hoc Networks," *Proceedings of the IEEE ICDCS 2001 Workshop on Wireless Networks and Mobile Computing,* pp. 413-418, Phoenix, AZ, April 2001.

[7] C.P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance Vector Routing (DSDV) for mobile Computers," *Proceedings of the ACM SIGCOMM'94,* pp. 234-244, September 1994.

[8] C-C. Chiang, H.-K. Wu, W. Liu, and M. Gerla, "Routing in Clustered Multihop Mobile Wireless Networks with Fading Channel," *Proceedings of the IEEE Singapore International Conference on Networks,* 1997.

[9] S. Murthy and J.J. Garcia-Luna-Aceves, "A Routing Protocol for Packet Radio Networks'," *Proceedings of the ACM First International Conference on Mobil Computing & Networking (MOBICOM' 95)*, November 1995.

[10] David B. Johnson and David A. Maltz, *Mobile Computing*, Kluwer Academic Publishers, 1996.

[11] V. Park and M. Scott Corson, "A Highly Adaptive Distributed Routing Algortihm for Mobile Wireless Networks," *Proceedings of the IEEE INFOCOM'97,* March 1996.

[12] C. Perkins and E. Royer, "Ad-Hoc On-Demand Distance Vector Routing," *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems Applications*, February 1999.

[13] C. Perkins and E. Royer, "Ad-Hoc On-Demand Distance Vector Routing," *Mobile Ad- Hoc Networking Working Group, INTERNET DRAFT, internet-drafts/draft-ietf-manet-aodv-11.txt,*, June 2002.

[14] Z. Haas and M. Pearlman, "Determining the Optimal Configuration of the Zone Routing Protocol," *IEEE Journal on Selected Areas in Communications*, Vol 17, No. 8, pp. 1395-1414, August 1999.

[15] Z. Haas and M. Pearlman, "The Zone Routing Protocol (ZRP) for Ad Hoc Networks," *INTERNET-DRAFT, internet-drafts/draft-ietf-manet-zone-zrp-04.txt,* July 2002.

[16] S. Basagni, "Distributed clustering for ad hoc networks," *Proceeding of the International Symposium on Parallel Architectures, Algorithms, and Networks (IS-PAN), Perth/Fremantle, Australia,* June 1999.

[17] A. B. McDonald, T. F. Znati, "Design and Performance of a Distributed Dynamic Clustering Algorithm for Ad-Hoc Networks," *Proceeding of the 34th Annual Simulation Symposium 2001*, 2001.

[18] T-C. Hou, T-J. Tsai, "An Access-Based Clustering Protocol for Multihop Wireless Ad- Hoc Networks," *IEEE Journal on Selected Areas in Communications,* Vol. 19, No. 7, pp. 1201-1210, July 2001.

[19] Stavros D. Nikolopoulos, Andreas Pitsillides, "Addressing Network Survivability Issues bye Finding the K-best Paths through a Trellis Graph," *Proceedings of the IEEE INFOCOM '97,* pp. 370-377, May 1997.

[20] A. Mawira, "Estimate of Mean C/I & Capacity of Interference Limited Mobile Ad-Hoc Network," *Proceedings of the IEEE International Zurich Seminar,* pp. 1-6, February 2002.