

**INSTITUTO TECNOLÓGICO Y DE ESTUDIOS
SUPERIORES DE MONTERREY
CAMPUS MONTERREY**

**Escuela de Tecnologías de Información y Electrónica
PROGRAMA DE GRADUADOS**



THESIS

**Tag Clustering A New Cooperative Algorithm For Topology
Organization in Ad-Hoc Networks**

BY

Ing. José Antonio Cervantes y Ramírez

MONTERREY, N. L.

DICIEMBRE 2005

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY

CAMPUS MONTERREY

Escuela de Tecnologías de Información y Electrónica
Programa de Graduados



Thesis

Tag Clustering A New Cooperative Algorithm For Topology Organization in Ad-Hoc
Networks

by

Ing. José Antonio Cervantes y Ramírez

Monterrey, N.L., Diciembre de 2005

© José Antonio Cervantes y Ramírez, 2005

Tag Clustering A New Cooperative Algorithm For Topology Organization in Ad-Hoc Networks

by

Ing. José Antonio Cervantes y Ramírez

Thesis

Presented to the graduate program of the

Escuela de Tecnologías de Información y Electrónica

as a partial fulfillment of the requirements for the degree of

Master of Science

major in

Telecommunications

Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Monterrey

Diciembre de 2005

This work is dedicated with love:
To my parents Benjamín Cervantes y Castro and María Antonieta Ramírez de
Cervantes. To my brother Juan Manuel Cervantes y Ramírez.

Acknowledgements

To my parents Benjamín Cervantes y Castro and María Antonieta Ramírez de Cervantes, for their unconditional support throughout my life. To my brother Juan Manuel Cervantes y Ramírez for his support at any moment.

I am very grateful to my thesis advisor César Vargas Rosales, Ph.D., for all his time during which we worked together, for all his professional advice for the accomplishment of this work and for his friendship. Because, without his guidance this thesis would not have been possible. Also I want to thank to José Ramón Rodríguez Cruz, Ph.D., and Jorge Carlos Mex Perera, Ph.D., for their valuable comments for the enhancement of this work.

To the ITESM and CONACYT for giving me the opportunity of realize my graduate studies.

I want to thank all my friends at CET, specially Pepe, Jeremías, Fernando, Paco, Victor, Fabian and Brenda. To my friends from Oaxaca Yalina, Eduardo, Gerardo, Gustavo, Ivan, Jose Luis and Himer for their friendship and helping at any moment.

JOSÉ ANTONIO CERVANTES Y RAMÍREZ

Instituto Tecnológico y de Estudios Superiores de Monterrey
Diciembre 2005

Contents

Acknowledgements	v
Abstract	vi
Abstract	vii
List of Tables	x
List of Figures	xi
Chapter 1 Introduction	1
1.1 Objective	2
1.2 Justification	2
1.3 Contribution	2
1.4 Organization	3
Chapter 2 Wireless Ad-Hoc Networks	4
2.1 Typical Operational Characteristics	4
2.2 Routing in Ad Hoc networks	6
2.3 Types of Ad Hoc Mobile Communications	7
2.4 Shortest Path Algorithms	8
2.4.1 Dijkstra Algorithm	8
2.5 Clustering, [1].	9
2.6 Multi-cluster network architecture and clustering Algorithm	11
2.6.1 Lowest-ID Cluster Algorithm	12
2.6.2 Lowest-ID heuristic	12
2.6.3 Highest-Connectivity Cluster Algorithm	13
2.6.4 Highest-Degree heuristic	14
2.6.5 Properties Of The Two Cluster Algorithms	15
Chapter 3 Model Description	16
3.1 Analogy with the Basketball Game	16
3.2 Statistics in the Tag Clustering	21

3.3	Tag clustering Algorithm	23
Chapter 4	Performance Evaluation	28
4.1	Proposed Scenarios	29
4.1.1	Scenario 1	30
4.1.2	Scenario 2	33
4.2	Finding the Shortest Path	38
4.3	Communication Overhead	39
4.4	Tag Clustering Algorithm vs Different Clustering Algorithms.	42
Chapter 5	Conclusion and Further Research	43
5.1	Conclusions	43
5.2	Further Research	44
	Bibliography	46
	Vita	48

List of Tables

4.1	Static Results for 3600 seconds	29
4.2	Static Results for different node density	31
4.3	Static Results for different communication range	34
4.4	Movement Results for different node density (speed 3 to 10 m/s)	35
4.5	Movement Results for different node density and speed	39
4.6	Summary of Different Clustering Algorithms	42

List of Figures

- 2.1 Ad hoc Wireless Network 5
- 2.2 At an airport, where people can access local- and wide-area networks, ad hoc Bluetooth connections are used to interconnect carried devices, such as PDAs, WCDMA mobile phones and notebook computers. For instance, a user might retrieve e-mail via a HiperLAN/2 interface to a notebook computer, but read messages and reply to them via his or her PDA. 7
- 2.3 Basic idea of Dijkstra’s algorithm. At the k^{th} step we have the set P of the k closest nodes to node 1 as well as the shortest distance D_i from node 1 to each node i in P . Of all paths connecting node 1 to some node not in P , the shortest one must pass exclusively through nodes in P (since $d_{ij} > 0$). 9
- 2.4 The Link-Clusterhead Architecture 10
- 2.5 Example of clustering 11
- 2.6 Example of cluster formation(lowest-ID) 12
- 2.7 Example of cluster formation(highest-connectivity) 14

- 3.1 Basketball Game 17
- 3.2 Clustering Ad-hoc network 17
- 3.3 Best basketball players and their statistics 19
- 3.4 Ideal Tag Clusterig 22
- 3.5 Formation of clusters in tag clustering algorithm 26
- 3.6 Tagging nodes in tag clustering algorithm 27

- 4.1 Empirical CDF for Single Nodes 32
- 4.2 Connectivity Weakness Probability 32
- 4.3 Connectivity Probability 33
- 4.4 Reachable clusters 34
- 4.5 Empirical CDF for Single Nodes 36
- 4.6 Connectivity Weakness Probability 36
- 4.7 Connectivity Probability 37
- 4.8 Reachable clusters 38

4.9	Shortest Path communication range 200m without clustering	40
4.10	Shortest Path communication range 150m without clustering	41
4.11	Shortest Path communication range 150m with tag clustering	41

Tag Clustering A New Cooperative Algorithm For Topology Organization in Ad-Hoc Networks

José Antonio Cervantes y Ramírez, M.Sc.
Instituto Tecnológico y de Estudios Superiores de Monterrey, 2005

Thesis advisor: Ph.D. César Vargas Rosales

A mobile ad hoc network is a dynamic mobile wireless network that can be formed without the need of any pre-existing wired or wireless infrastructure. One of the main challenges in an ad hoc network is the design of robust topology organization algorithms that adapt to the frequent and randomly changing network topology. Organizing mobile nodes into manageable clusters can reduce routing overhead and provide more scalable solutions. In this thesis, we propose a tag clustering architecture for topology organization in mobile ad hoc networks. Using this clustering architecture, hybrid routing schemes can be employed for intra-cluster and inter-cluster routing to improve the performance of routing. All nodes use different statistics to perform routing and clustering operations. They collect routing and clustering information and periodically maintain the corresponding tables. We identify various parameters to improve network performance in the proposed clustering architecture. By using some optimal values for these clustering and routing parameters, more robust and scalable routing can be attained.

Tag Clustering A New Cooperative Algorithm For Topology Organization in Ad-Hoc Networks

José Antonio Cervantes y Ramírez, M.Sc.
Instituto Tecnológico y de Estudios Superiores de Monterrey, 2005

Thesis advisor: Ph.D. César Vargas Rosales

Una red ad hoc es una red inalámbrica móvil que puede ser formada sin la necesidad de una red de infraestructura alámbrica o inalámbrica pre-existente. Uno de los mayores retos en las redes ad hoc es el diseño de algoritmos de organización de topología robustos que se adapten a los cambios aleatorios frecuentes de la topología de la red. Organizar a los nodos móviles en clusters manejables puede reducir el overhead del ruteo y proveer soluciones mas escalables. En esta tesis se propone una arquitectura tipo tag clustering para organización de la topología en las redes ad hoc. Utilizando esta arquitectura de clustering, esquemas híbridos pueden ser utilizados para un ruteo intra-cluster y inter-cluster para mejorar el funcionamiento del ruteo. Todos los nodos utilizan diferentes estadísticas para el funcionamiento del ruteo y las operaciones del clustering. Se recolecta informacion de ruteo y de clustering y periódicamente se mantienen las tablas correspondientes. Se identifican varios parámetros para mejorar el funcionamiento de la red en la arquitectura de clustering propuesta. Utilizando algunos valores para los parámetros del clustering y del ruteo, se puede alcanzar un ruteo mas robusto y escalable.

Chapter 1

Introduction

Ad hoc wireless networks consist of mobile nodes interconnected by multihop communication paths. Unlike conventional wireless networks, ad hoc networks have no fixed network infrastructure or administrative support. The topology of the network changes dynamically as mobile nodes join or depart the network or radio links between nodes become unusable. Ad hoc wireless networks are self-creating, self-organizing, and self-administering. They come into being solely by interactions among their constituent wireless mobile nodes, and only such interactions are used to provide the necessary control and administration functions supporting such networks.

The ad hoc wireless networks offer unique benefits and versatility for certain environments and certain applications. No preexisting fixed infrastructure, including base stations, being prerequisite, they can be created and used “any time, anywhere.” Second, such networks could be intrinsically fault-resilient, for they do not operate under the limitations of a fixed topology. Indeed, since all nodes are allowed to be mobile, the composition of such networks is necessarily time-varying. Addition and deletion of nodes occur only by interactions with other nodes; no other agency is involved. Such perceived advantages elicited immediate interest in the early days among military, police, and rescue agencies in the use of such networks, especially under disorganized or hostile environments, including isolated scenes of natural disaster and armed conflict. In recent days, home or small office networking and collaborative computing with laptop computers in a small area (e.g., a conference or classroom, single building, convention center) have emerged as other major areas of potential application. In addition, people also recognize that ad hoc networking has obvious potential application in all the traditional areas of interest for mobile computing.

Numerous challenges must be overcome to realize the practical benefits of ad hoc networking. These include effective routing, medium (or channel) access, mobility management, power management, security, and quality of service (QoS) issues, mainly pertaining to delay and bandwidth management. Cost-effective resolution of these issues at appropriate levels is essential for widespread general use of ad hoc networking.

The absence of fixed infrastructure means that the nodes of an ad hoc network communicate directly with one another in a peer-to-peer fashion. The mobility of these nodes imposes limitations on their power capacity, and hence on their transmission range; indeed, these nodes often must satisfy stringent weight limitations for portability. Assuming ubiquitous IP networking as the underlying model for our discussion, it is evident that each node must therefore be able to function as a router as well. As the nodes move in and out of range with respect to other nodes, including those operating as routers, the instantaneous topology changes must somehow be communicated to all other nodes as appropriate. In accommodating the communication needs of the user applications, the limited bandwidth of wireless channels and their generally hostile transmission characteristics impose additional constraints on how much administrative and control information may be exchanged, and how often. Ensuring effective routing is one of the great challenges for ad hoc networking,[13].

1.1 Objective

The objective of this thesis is to develop a cooperative topology organization algorithm based on clustering concepts to allow network topology control and to evaluate the performance of ad hoc networks with respect to the organization, connectivity and mobility of the nodes, using the statistics proposed in the tag clustering algorithm.

1.2 Justification

The ad hoc networks are systems of fast implementation that do not count on permanent physical infrastructure. Therefore, it is recovery to obtain the interconnections of nodes source and destination, and necessary to establish topologies on which the routes are constructed that will transport the information. However, given the ample class of stations (fixed, semi-portable, mobiles), it is necessary to study the way in which the nodes will be organized to be able to define a topology, and the necessary connectivity for its operation, formation and maintenance.

1.3 Contribution

We propose a new algorithm for ad-hoc networks, with the introduced concepts of collaborative work with the purpose of improving organization, connectivity and mobility of ad-hoc networks.

1.4 Organization

This work is organized as follows. In Chapter 2, we present an overview of ad hoc networks. In this chapter we detail some ad hoc routing protocols and provide an overview of clustering. Chapter 3 presents the model and the analogy with the concepts of cooperative work in the basketball game. Chapter 4 shows numerical results obtained from simulations. Finally, Chapter 5 presents conclusions and future work.

Chapter 2

Wireless Ad-Hoc Networks

A mobile ad hoc network is a network formed without any central administration which consists of mobile nodes that use a wireless interface to send packet data. Since the nodes in a network of this kind can serve as routers and hosts, they can forward packets on behalf of other nodes and run user applications. As we can see in Figure 2.1

In contrast to traditional wire-line or wireless networks, an ad hoc network could be expected to operate in a network environment in which some or all the nodes are mobile.

In this dynamic environment, the network functions must run in a distributed fashion, since nodes might suddenly disappear from, or show up in, the network. In general, however, the same basic user requirements for connectivity and traffic delivery that apply to traditional networks will apply to ad hoc networks,[9].

2.1 Typical Operational Characteristics

Distributed operation: a node in an ad hoc network cannot rely on a network in the background to support security and routing functions. Instead these functions must be designed so that they can operate efficiently under distributed conditions.

Dynamic network topology: in general, the nodes will be mobile, which sooner or later will result in a varying network topology. Nonetheless, connectivity in the network should be maintained to allow applications and services to operate undisturbed. In particular, this will influence the design of routing protocols. Moreover, a user in the ad hoc network will also require access to a fixed network (such as the Internet) even if nodes are moving around. This calls for mobility management functions that allow network access for devices located several radio hops away from a network access point.

Fluctuating link capacity: the effects of high bit-error rates might be more pro-

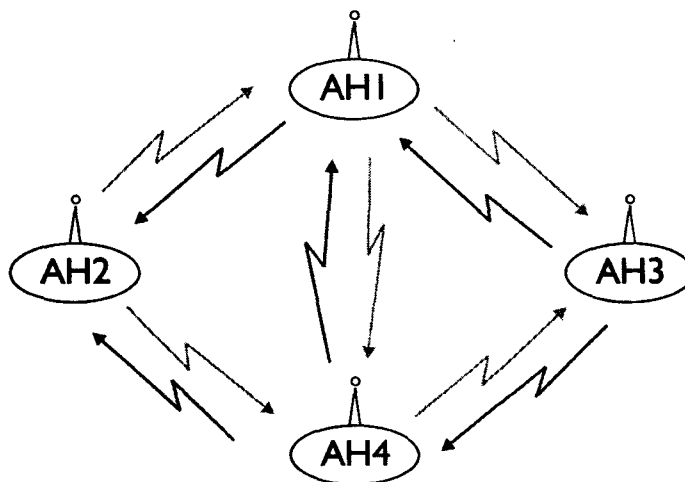


Figure 2.1: Ad hoc Wireless Network

found in a multihop ad hoc network, since the aggregate of all link errors is what affects a multihop path. In addition, more than one end-to-end path can use a given link, which if the link were to break, could disrupt several sessions during periods of high bit-error transmission rates. Here, too, the routing function is affected, but efficient functions for link layer protection (such as forward error correction, FEC, and automatic repeat request, ARQ) can substantially improve the link quality.

Low-power devices: in many cases, the network nodes will be battery-driven, which will make the power budget tight for all the power-consuming components in a device. This will affect, for instance, CPU processing, memory size/usage, signal processing, and transceiver output/input power. The communication-related functions (basically the entire protocol stack below the applications) directly burden the application and services running in the device. Thus, the algorithms and mechanisms that implement the networking functions should be optimized for lean power consumption, so as to save capacity for the applications while still providing good communication performance.

Besides achieving reasonable network connectivity by using multihop routes, the introduction of multiple radio hops might also improve overall performance, given a constrained power budget. Today, however, this can only be realized at the price of more complex routing,[9].

The inherent stochastic communications quality in a wireless ad hoc network, makes it difficult to offer fixed guarantees on the services offered to a device. In networks of this kind, fixed guarantees would result in requirements for how nodes move, as well as requirements for node density, which would inherently inhibit the notion of ad hoc

operation.

To further improve user perception of the service, user applications that run over an ad hoc network could be made to adapt to sudden changes in transmission quality. QoS support in an ad hoc network will affect most of the networking functions, especially routing and mobility. In addition, local buffer management and priority mechanisms must be deployed in the devices in order to handle differentiated traffic streams,[9].

2.2 Routing in Ad Hoc networks

For mobile ad hoc networks, the issue of routing packets between any pair of nodes becomes a challenging task because the nodes can move randomly within the network. A path that was considered optimal at a given point in time might not work at all a few moments later. Moreover, the stochastic properties of the wireless channels add to the uncertainty of path quality. The operating environment as such might also cause problems for indoor scenarios-the closing of a door might cause a path to be disrupted.

Several routing protocols have been proposed for these networks recently. These routing protocols can be classified into three main categories: proactive, reactive and hybrid.

Traditional routing protocols are proactive in that they maintain routes to all nodes, including nodes to which no packets are being sent. They react to any change in the topology even if no traffic is affected by the change, and they require periodic control messages to maintain routes to every node in the network. The rate at which these control messages are sent must reflect the dynamics of the network in order to maintain valid routes. Thus, scarce resources such as power and link bandwidth will be used more frequently for control traffic as node mobility increases,[9].

An alternative approach involves establishing reactive routes, which dictates that routes between nodes are determined solely when they are explicitly needed to route packets. This prevents the nodes from updating every possible route in the network, and instead allows them to focus either on routes that are being used, or on routes that are in the process of being set up.

The hybrid protocols use features of both reactive and proactive protocols. Non-clustered routing protocols have a flat routing architecture where each mobile node maintains complete routing information. Therefore they have routing overhead, which increases considerably with the network size. The main advantage of hybrid protocols is their flexibility in allowing the use of different routing mechanisms within and be-

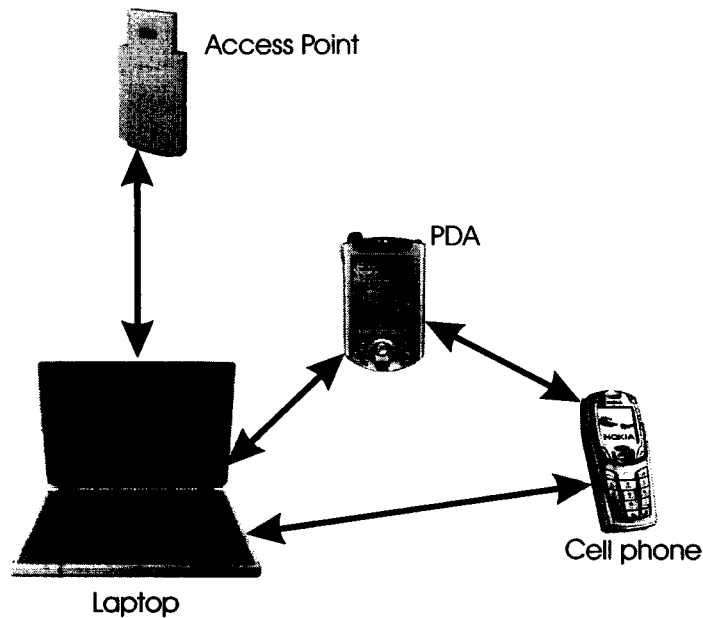


Figure 2.2: At an airport, where people can access local- and wide-area networks, ad hoc Bluetooth connections are used to interconnect carried devices, such as PDAs, WCDMA mobile phones and notebook computers. For instance, a user might retrieve e-mail via a HiperLAN/2 interface to a notebook computer, but read messages and reply to them via his or her PDA.

tween clusters. Several routing protocols based on clustering architecture have been proposed in recent years. Organizing clusters hierarchically reduces routing overhead and increases scalability,[6].

2.3 Types of Ad Hoc Mobile Communications

Ad-hoc networking covers multihop scenarios, where network nodes communicate via other network nodes, such as conference, hospital, battlefield, rescue, and monitoring scenarios. Particular ad hoc networks systems include packet radio networks, sensor networks, personal communication systems, rooftop (mesh) networks for wireless internet access, and wireless local area networks. Ad hoc networks may consist of self-organized standalone nodes, or can be linked to a fixed infrastructure via access point (such networks are also called hybrid ad hoc networks),[5].

Today, many people carry numerous portable devices, such as laptops, mobile phones, PDAs and mp3 players, for use in their professional and private lives. For the most part, these devices are used separately-that is, their applications do not interact. Imagine, however, if they could interact directly: participants at a meeting could share doc-

uments or presentations; business cards would automatically find their way into the address register on a laptop and the number register on a mobile phone; as commuters exit a train, their laptops could remain online; likewise, incoming email could now be diverted to their PDAs; finally, as they enter the office, all communication could automatically be routed through the wireless corporate campus network.

Today, our vision of ad hoc networking includes scenarios such as those depicted in Figure 2.2, where people carry devices that can network on an ad hoc basis. A users devices can both interconnect with one another and connect to local information points for example, to retrieve updates on flight departures, gate changes, and so on. The ad hoc devices can also relay traffic between devices that are out of range. The airport scenario thus contains a mixture of single and multiple radio hops.

These examples of spontaneous, ad hoc wireless communication between devices might be loosely defined as a scheme, often referred to as ad hoc networking, which allows devices to establish communication, anytime and anywhere without the aid of a central infrastructure. Actually, ad hoc networking as such is not new, but the setting, usage and players are. In the past, the notion of ad hoc networks was often associated with communication on combat fields and at the site of a disaster area; the scenario of ad hoc networking is likely to change, as is its importance,[9].

2.4 Shortest Path Algorithms

Due to the nature of routing applications, we need flexible and efficient shortest path procedures, both from a processing time point of view and also in terms of the memory requirements. Shortest Path problems are among the most studied network flow optimization problems, with interesting applications in a range of fields. One such application is in the field of ad hoc networks routing systems. Conventional techniques for solving shortest paths routing algorithms have been based on Dijkstras Shortest Path algorithm.

2.4.1 Dijkstra Algorithm

This algorithm requires that all arc lengths be positive (fortunately, the case for data networks applications). The general idea is to find the shortest paths in order to increasing path length. The shortest of the shortest paths from node 1 must be the single-arc path to the closest neighbor of node 1, since any multiple-arc path must be longer than the first arc length because of the positive length assumption. The next shortest paths must either be the single-arc path to the next closest neighbor of 1 or the shortest two-arc path through the previously chosen node, etc. To formalize this

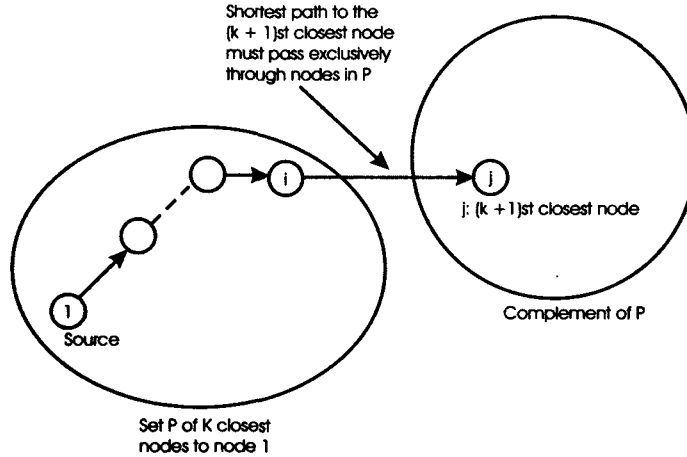


Figure 2.3: Basic idea of Dijkstra's algorithm. At the k^{th} step we have the set P of the k closest nodes to node 1 as well as the shortest distance D_i from node 1 to each node i in P . Of all paths connecting node 1 to some node not in P , the shortest one must pass exclusively through nodes in P (since $d_{ij} > 0$).

procedure into an algorithm, we view each node i as being labeled with an estimate D_i of the shortest path length from node 1. When the estimate becomes certain, we regard the node as being permanently labeled, and keep track of this with a set P of permanently labeled nodes. The node added to P at each step will be the closest to node 1 out of those that are not yet in P . Figure 2.3 illustrates the main idea. The detailed algorithm is as follows,[4]:

Initially $P = \{1\}$, $D_1 = 0$, and $D_j = d_{1j}$ for $j \neq 1$.

Step 1. (Find the next closest node.) Find $i \in P$ such that

$$D_i = \min_{j \in P} D_j \tag{2.1}$$

Set $P := P \cup \{i\}$. If P contains all node then stop; the algorithm is complete.

Step 2. (Updating of labels.) For all $j \in P$ set

$$D_j := \min_i [D_j, D_i + d_{ij}] \tag{2.2}$$

Go to Step 1.

2.5 Clustering, [1].

Mobile ad hoc networks for many applications are naturally clustered by geographical or physical boundaries A cluster is a collection of interconnected groups of

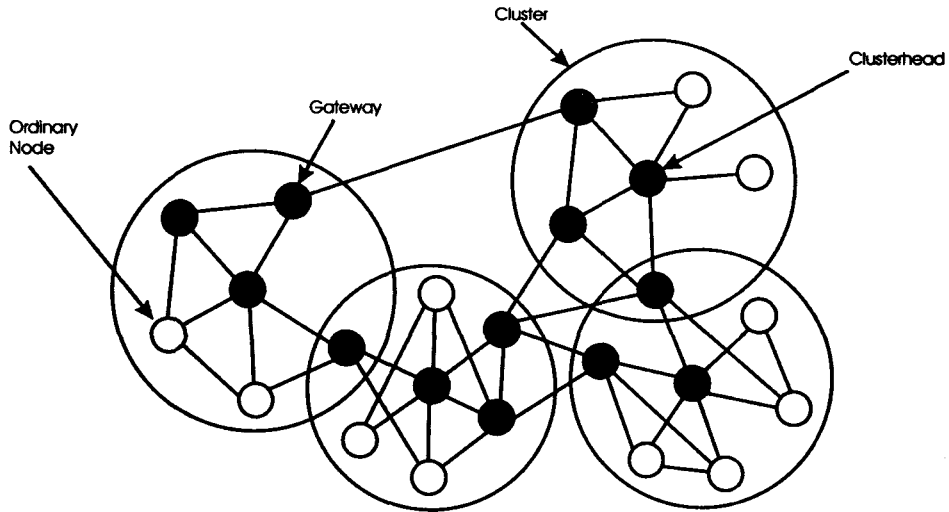


Figure 2.4: The Link-Clusterhead Architecture

nodes that cover all mobile nodes in the network. Clusters can be either distinct or overlapping. In the former, each node belongs to only one cluster while in the latter, the neighboring clusters can have a common node (gateway) between them,[6].

Each cluster contains a clusterhead, one or more gateways, and zero or more ordinary nodes that are neither clusterheads nor gateways as shown in Figure 2.4. The clusterhead schedules transmissions and allocates resources within the clusters. Gateways connect adjacent clusters. A gateway may directly connect two clusters by acting as a member of both, or it may indirectly connect two clusters by acting as a member of one and forming a link to a member of the other,[12].

Most hierarchical clustering architectures for mobile radio networks are based on the concept of clusterhead. The clusterhead acts as a local coordinator of transmissions within the cluster. It differs from the base station concept in current cellular systems, in that it does not have special hardware and in fact is dynamically selected among the set of stations. However, it does extra work with respect to ordinary stations, and therefore it may become the bottleneck of the cluster.

Clustering offers several advantages in mobile ad hoc networks. First, network partitioning improves routing and mobility management,[14]. It increases system capacity, reduces signaling and control overhead and minimizes network congestion. This makes the network more scalable and enables support for larger network sizes. Second, clustering stabilizes the network topology and provides a virtual infrastructure for a dynamic network. Here, the clusterhead acts as a base station for its cluster. Third, clustering helps to perform more efficient resource allocation. By assigning different codes to each

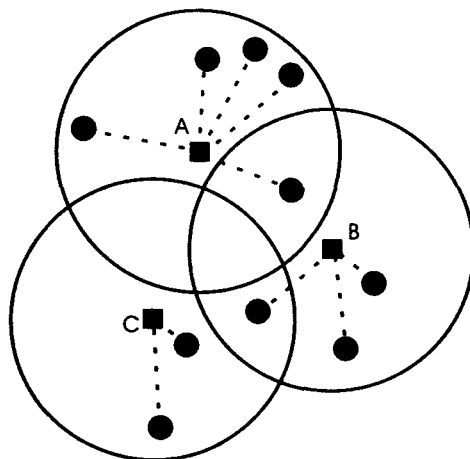


Figure 2.5: Example of clustering

cluster, MAC resource management can be improved and the wireless channel can be used efficiently,[7],[3].

2.6 Multi-cluster network architecture and clustering Algorithm

Clustering provides a convenient framework for the development of important features such as code separation (among clusters), channel access, routing, power control, virtual circuit support and bandwidth allocation. Following the clustering approach, the entire population of nodes is grouped into clusters. A cluster is a subset of nodes which can (two-way) communicate with a clusterhead and (possibly) with each other. In Figure 2.6, nodes A, B and C are clusterheads for their coverage area respectively. Each of them serves as a regional broadcast node, and as a local coordinator to enhance channel throughput. Within a cluster, we can easily enforce time-division scheduling. Across clusters, we can facilitate spatial reuse of time slots and codes. The objective of the clustering algorithm is to find a feasible interconnected set of clusters covering the entire node population. A good clustering algorithm should be stable to radio motion, i.e. it should not change the cluster configuration too drastically when a few nodes are moving and the topology is slowly changing. Otherwise, the clusterheads will not control their clusters efficiently and thus lose their role as local coordinators. To this end, two distributed clustering algorithms are considered. One is the lowest-ID algorithm [2]: the lowest-ID node in a neighborhood is elected as the clusterhead. The other is the highest-connectivity (degree) algorithm, which is a modified version of [11]. In this case, the highest degree node in a neighborhood becomes the clusterhead, [7].

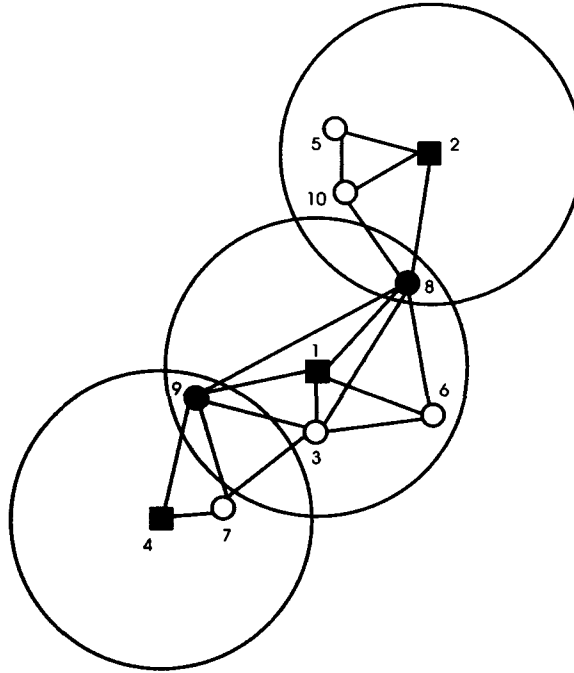


Figure 2.6: Example of cluster formation (lowest-ID)

2.6.1 Lowest-ID Cluster Algorithm

Each node is assigned a distinct ID. Periodically, the node broadcasts the list of nodes that it can hear (including itself).

- A node which only hears nodes with ID higher than itself is a “clusterhead” (CH).
- The lowest-ID node that a node hears is its clusterhead, unless the lowest-ID specifically gives up its role as a clusterhead (deferring to a yet lower ID node).
- A node which can hear two or more clusterheads is a “gateway”.
- Otherwise, a node is an ordinary node.

Figure 2.6 shows a 10 node example, where nodes 1, 2 and 4 are clusterheads; nodes 8, 9 are gateway nodes, [7].

2.6.2 Lowest-ID heuristic

The Lowest-ID, also as known as identifier-based clustering, was originally proposed by Baker and Ephremides [2]. This heuristic assigns a unique id to each node and chooses the node with the minimum id as a clusterhead. Thus, the ids of the neighbors of the clusterhead will be higher than that of the clusterhead. However, the clusterhead can delegate its responsibility to the next node with the minimum id in its cluster. A

node is called a gateway if it lies within the transmission range of two or more clusterheads. Gateway nodes are generally used for routing between clusters. Only gateway nodes can listen to the different nodes of the overlapping clusters that they lie. The concept of distributed gateway (DG) is also used for inter-cluster communication only when the clusters are not overlapping. DG is a pair of nodes that lies in different clusters but they are within the transmission range of each other. The main advantage of distributed gateway is maintaining connectivity in situations where any clustering algorithm fails to provide connectivity. For this heuristic, the system performance is better compared with the Highest-Degree heuristic in terms of throughput. Since the environment under consideration is mobile, it is unlikely that node degrees remain stable resulting in frequent clusterhead updates. However, the drawback of this heuristic is its bias towards nodes with smaller ids which may lead to the battery drainage of certain nodes. One might think that this problem may be fixed by renumbering the node ids from time to time, which is however non-trivial. There are other problems associated with such renumbering. For instance, the optimal frequency of renumbering would need to be determined so that the system performance is maximized. More importantly, every time node ids are reshuffled, the neighboring list of all the nodes need also to be changed. If we consider that the nodes are numbered in the increasing order of their remaining battery power, then a centralized algorithm is required. We can avoid this by exchanging ids between nodes and making sure that the uniqueness of ids are maintained. Even then, the clustering has to be redone which would add unnecessary computational complexity to the system. For example, suppose two nodes mutually exchange their ids in order to keep the ids according to their remaining battery power. After exchanging, all nodes that were connected to these two nodes, regardless of their status (clusterhead or ordinary node), need to be notified of the change so that they can update their neighbor list. This effect may propagate and add overhead to the system. Moreover, it does not attempt to balance the load uniformly across all the nodes, [10].

2.6.3 Highest-Connectivity Cluster Algorithm

Each node broadcasts the list of nodes that it can hear (including itself).

- A node is elected as a clusterhead if it is the most highly connected node of all its “uncovered” neighbor nodes (in case of a tie, lowest ID prevails).
- A node which has not elected its clusterhead yet is an “uncovered” node, otherwise it is a “covered” node.
- A node which has already elected another node as its clusterhead gives up its role as a clusterhead.

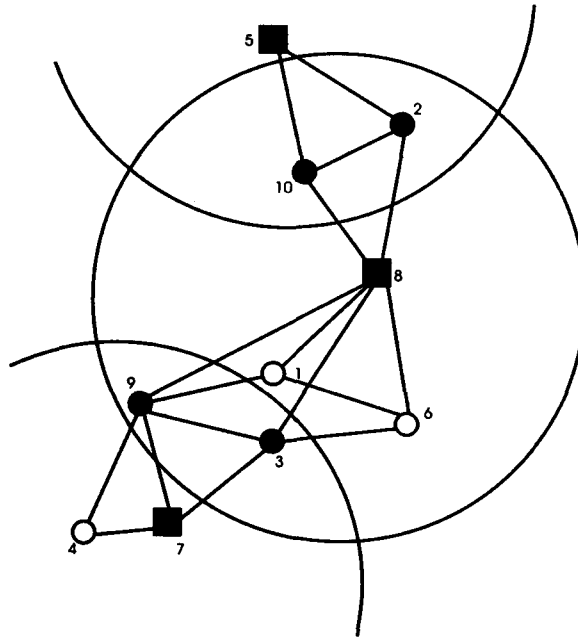


Figure 2.7: Example of cluster formation(highest-connectivity)

Figure 2.7 shows the same 10 node example, where nodes 5,7 and 8 are cluster-heads; nodes 2, 3, 9, 10 are gateway nodes, [7].

2.6.4 Highest-Degree heuristic

The Highest-Degree, also known as connectivity-based clustering, was originally proposed by Gerla and Parekh [11] in which the degree of a node is computed based on its distance from others. Each node broadcasts its id to the nodes that are within its transmission range. A node x is considered to be a neighbor of another node y if x lies within the transmission range of y . The node with maximum number of neighbors (i.e., maximum degree) is chosen as a clusterhead and any tie is broken by the unique node ids. The neighbors of a clusterhead become members of that cluster and can no longer participate in the election process. Since no clusterheads are directly linked, only one clusterhead is allowed per cluster. Any two nodes in a cluster are at most two-hops away since the clusterhead is directly linked to each of its neighbors in the cluster. Basically, each node either becomes a clusterhead or remains an ordinary node (neighbor of a clusterhead). Experiments demonstrate that the system has a low rate of clusterhead change but the throughput is low under the Highest-Degree heuristic. Typically, each cluster is assigned some resources which is shared among the members of that cluster on a round-robin basis [11]. As the number of nodes in a cluster is increased, the throughput drops and hence a gradual degradation in the system performance is observed. The reaffiliation count of nodes are high due to node movements and as a

result, the highest-degree node (the current clusterhead) may not be re-elected to be a clusterhead even if it loses one neighbor. All these drawbacks occur because this approach does not have any restriction on the upper bound on the number of nodes in a cluster, [10].

2.6.5 Properties Of The Two Cluster Algorithms

Following these properties of clustering algorithms, each node is either a clusterhead itself or is directly linked to one or more clusterheads. Note that property (a) allows only one clusterhead per cluster. The clustering algorithm must be performed as rapidly as possible, so that each clusterhead can take and maintain control of its members efficiently, [7].

- (a) No cluster heads are directly linked.
- (b) In a cluster, any two nodes are at most two-hops away, since the clusterhead is directly linked to every other node in the cluster.

In this chapter we presented an introduction to ad hoc networks and discussed the general operating principles of the clustering algorithms as well as the properties that are desirable in ad hoc networks. Nowadays, there are collaborative works that are focused on the transmission area, coding, information processing, but not related to topology generation, organization and maintenance. With this information we can propose a new cooperative algorithm for topology organization in ad hoc networks.

Chapter 3

Model Description

To improve the performance of routing protocols, the clustering architecture should be stable. Moreover, the storage, computation and communication overhead at each node should be relatively low. To achieve these goals, optimal values for the clustering parameters are required. Since a single parameter is not sufficient to attain this, an aggregate clustering metric is necessary.

An aggregate clustering metric that includes node mobility and link quality history as well as available bandwidth can be used to apply a predictive mechanism to stabilize the clustering architecture. This metric is maintained at the clusterhead and will be used to assess the quality of the clusterhead periodically. The mobile agent examines this metric before deciding whether a new clusterhead is to be elected or not. This procedure avoids the loss of routing and clustering packets when the clusterhead fails,[6].

3.1 Analogy with the Basketball Game

Please notice that if it were possible to watch a basketball game from above we could see there is an analogy with ad-hoc networks, that is because the mobility of the players, the interchange of the ball within them and the score. For example lets see Figure 3.1, in this case we have a typical basketball game, the black team wants to score, and the white team defends its basket. Each player can communicate with each other or of the white or black team.

Now, in Figure 3.2, we show a cluster topology for ad hoc networks. We notice a similarity between the two figures. For this, we can introduce a new cluster algorithm that has some characteristics of the game of basketball based on the teamwork of the players. Next, we explain this analogy an the meaning in our model.

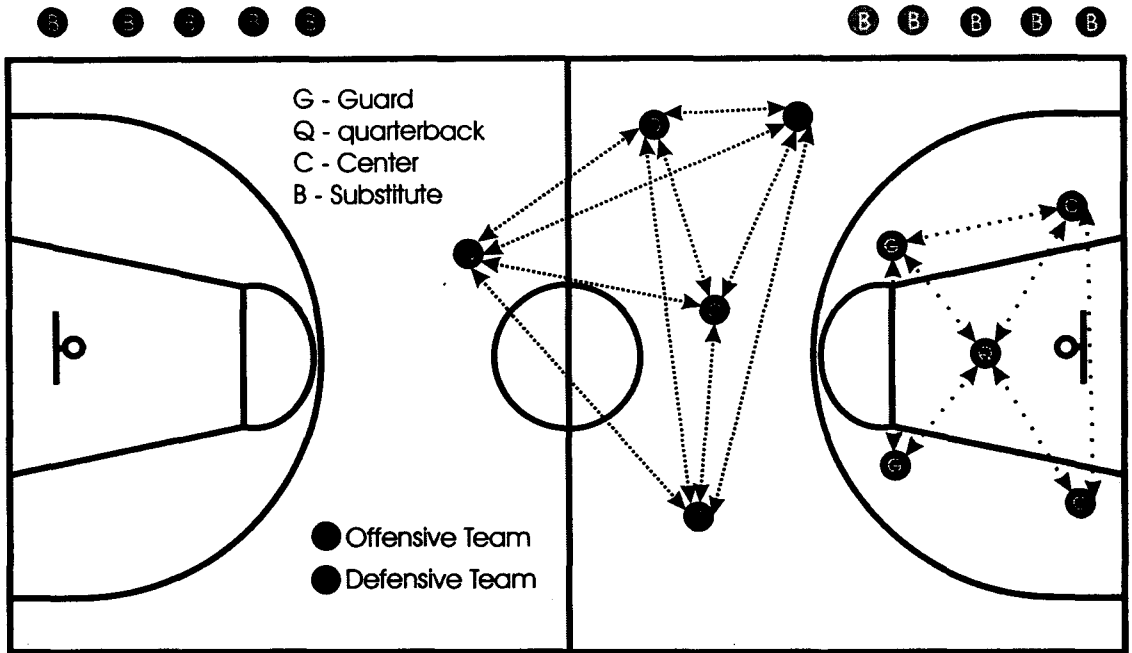


Figure 3.1: Basketball Game

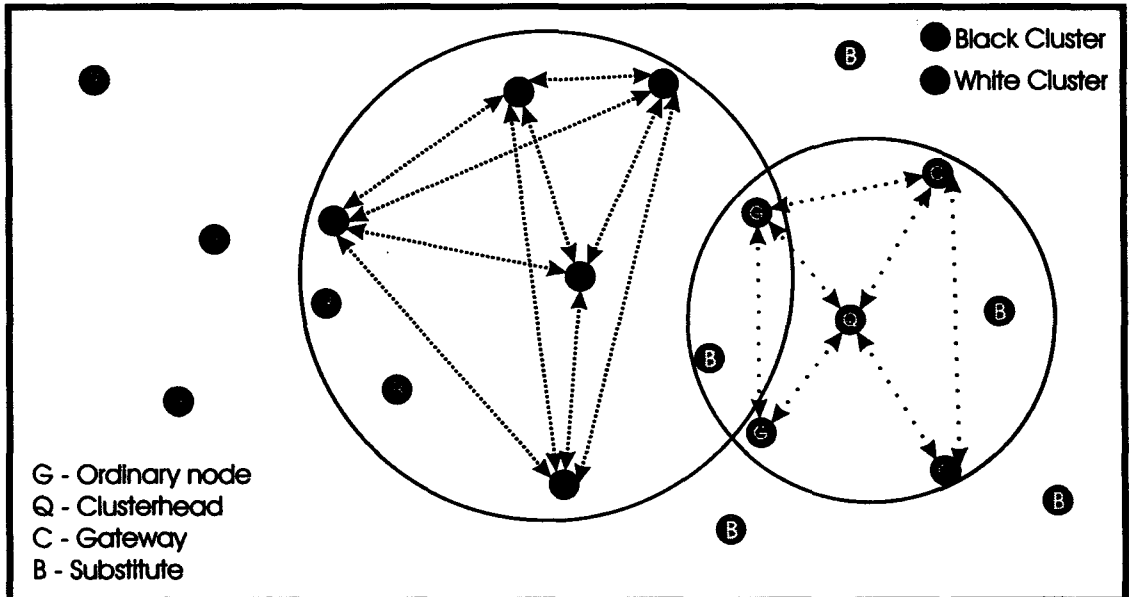


Figure 3.2: Clustering Ad-hoc network

Basketball game (definition):

Basketball is played by two (2) teams of five (5) players each. The aim of each team is to score in the opponents' basket and to prevent the other team from scoring. Basketball is controlled by officials, table officials and a commissioner.

Tag Clustering:

A cluster will be conformed by a team of a number of nodes each. The aim of each cluster is to generate routes to interchange packets of information between the nodes. Each equipment will be conformed by clusterhead (quarterback), ordinary nodes (guards and benches) and Gateways (Centers).

Basketball game (players):

Five (5) players from each team shall be on the court during playing time and may be substituted. One quarterback, two guards and two centers.

The quarterback is who organized the play, it's faster than the other players and passes the ball in several occasions. The guard is who score the majority of the points. The center is the tallest player and he takes the defensive rebounds.

A substitute becomes a player and a player becomes a substitute when: The official beckons the substitute to enter the playing court. During a charged time-out or an interval of play, a substitute requests the substitution to the scorekeeper.

We take the statistics¹ of the best player of all the times in the NBA as we show in Figure 3.3

Tag Clustering:

The Clusterhead (quarterback) will be in charge to direct the team. Within the statistics it will be that node that has the greater number of participation to reveal packtes to another node destination (APG). Its trustworthiness in the connection and its proximity with other clusters will be greater than the guards but smaller than the center's.

¹PPG= Points per Game, RPG= Rebound per Game, APG= Assists per Game.

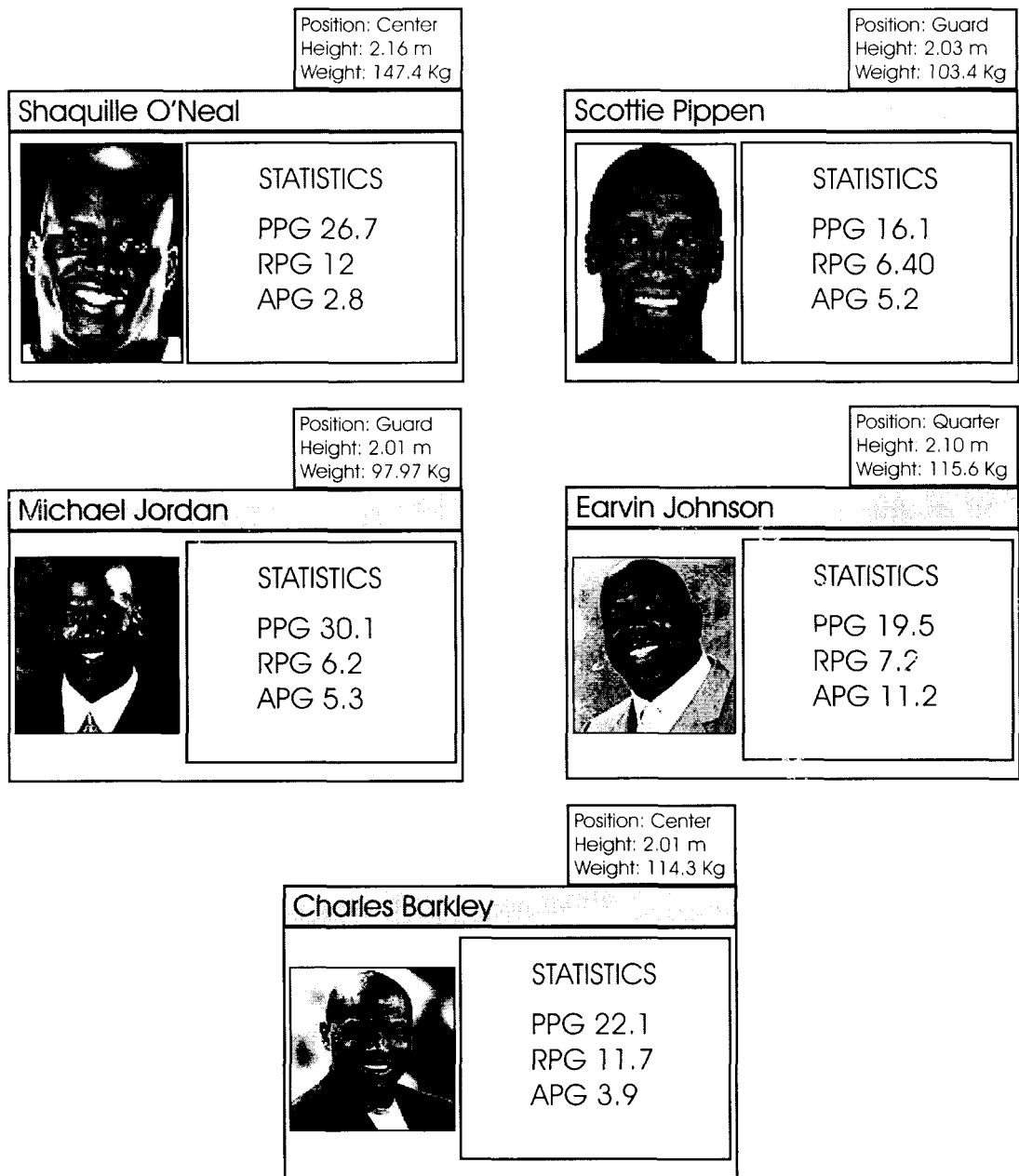


Figure 3.3: Best basketball players and their statistics

It will be in charge to take the statistics of the team (if sizing is not used). It will be labeled with MX where X is the number of the cluster to which it belongs.

The guards will be those nodes that have the greater number of packets transmitted by time unit (PPG). Their trustworthiness in the connection and its proximity with other clusters will be similar. They will be labeled with GX where X is the number of the equipment to which they belongs.

The centers will be those nodes that have the greater number of rescue of a route, i.e., if some route is broken during the transmission, these nodes participate to rescue the transmission of the package (RPG). They have a trustworthiness in the connection and a proximity with other clusters much greater than the clusterhead and the guards. They will be labeled with CX where X is the number of the cluster to which they belongs.

The substitute will be conformed by those nodes that have two or more different connections with clusters, will be up to three jumps of distance from the gateway and can belong to two or more clusters, i.e., we consider overlapped clusters. The decision to belong to a cluster or to enter one of the clusters will be the responsibility of the clusterhead. They will be labeled with BX, where X is the number of the cluster to which it belongs. Also we can have substitutes in the ordinary nodes for the centers and for the clusterhead, we do not consider substitutes for the guards.

Basketball game (fouls):

During a game in which ten (10) players are moving at high speed in a limited space, personal contact cannot be avoided. A foul is an infraction of the rules concerning illegal personal contact with an opponent and/or unsportsmanlike behavior. Any number of fouls may be called against a team. Irrespective of the penalty, each foul shall be charged, entered on the score sheet against the offender and penalized accordingly. If a player commits five fouls is expelled from the game.

Tag Clustering:

Fouls are the routes that have disappeared by the birth or death of a node, by a coverage anomaly in the transmission of a node, labeled interchange of or some other situation that affects a trajectory of communication between the nodes. When accumulating 5 fouls the node will be labeled like nonreliable and a timer will be start to

count for 5 minutes, and until the term of 5 minutes it could again be considered for the generation of routes. It will be labeled with FX where X is the number of the cluster to which it belongs.

The markers of fouts of the nodes, must be to available to the nodes to generate reliable routes. When accumulating 5 fouts a node will be labeled like nonreliable. When accumulating 5 foutes by equipment, is to say that the equipment are accumulated 25 fouts (each one of the 5 members has fouts in its account) the equipment is not reliable and cluster will not be taken into account for outer routes to the equipment including all the cluster.

The death of a node is a random event when, the node is no longer present in the network, or in the case where a node due to mobility lies outside a cluster which used to be part of. Also, the low trustworthiness in the connection causes a node to die. The clusterhead registers this information in the statistics. A dead node will be labeled with DX, where X is the number of the cluster to which it belongs.

The dead time is the calculation of the inactivity of some node within a cluster, i.e., the node does not participate when commanding messages or like part of a route. The dead times will be defined within the tables of statistics.

The birth of a node is a random event when a new node appears within a cluster and it will have to identify itself with the clusterhead and after passing the 5 minutes labeled as unreliable, the corresponding label will be granted to him. It will be labeled with NX, where X is the number of the equipment to which it belongs.

We show the tag of every node in Figure 3.4, in the next section we talk about the statistics used in the tag of the nodes.

3.2 Statistics in the Tag Clustering

Each node has a routing table that indicates the possible trajectories, the statistics and the label of each attainable node. The control table contains all the located statistics of clusters within its coverage area.

A signal is sent to all the members of the cluster to indicate that the statistics are reset. This reset of statistics does not change the node's label. The restart takes control of trustworthiness aims of the routes, since the constant movement of the nodes and the birth and death of nodes affect the generation of the routes. With the restart,

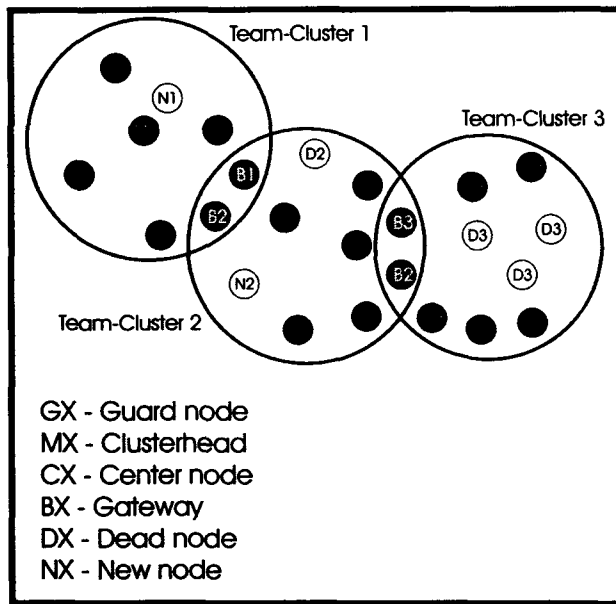


Figure 3.4: Ideal Tag Clusterig

the nodes try to diminish this aspect.

The control table will be part of the routing table, i.e., within the table, we find measures of efficiency of each node in the cluster, these measures will be handled by the clusterhead. The clusterhead has knowledge of all the nodes in the cluster, so that, it can make a decision about which route is more efficient to be followed.

The statistics must be provided by the clusterhead and must be available to the nodes for routing decisions. Within the cluster, routes must have clearly visible statistics for all the involved nodes in the network, including others from different clusters. The statistics must indicate, at least:

- Identification of the cluster.
- Trustworthiness in the connection and proximity to other clusters.
- Identification of the nodes who participate within the cluster.
- Identification of the nodes that have gateways (centers) as well as the neighboring clusters that can be reached.
- Accumulated fouls by each node.

- Accumulated fouls by each cluster.
- Packets transmitted by time unit (PPG) by node.
- Number of participation to reveal packets to another node destination (APG).
- Rescue of routes (RPG) by each node.
- The routes marked by each cluster and, preferably the routes in which each node participates.
- Active routes with time label.

3.3 Tag clustering Algorithm

Tag Clustering Algorithm is a clustering algorithm designed to use in mobile ad hoc networks. The algorithm divides the nodes of the ad hoc network into a number of overlapping or disjoint 1-hop- diameter clusters in a distributed manner. A cluster head (quarter) is elected for each cluster to maintain cluster statistic information. Inter-cluster routes are discovered dynamically using the cluster player's tags information kept at each cluster head. By clustering nodes into teams, the algorithm efficiently minimizes the number of gateways (centers) and broken routes could be repaired locally without rediscovery with the substitutes using cooperative work. The objective of the tag clustering algorithm is to find a feasible interconnected set of clusters covering the entire node population.

Tag clustering algorithm has two logical stages: first, the formation of clusters and, second, the tagging of the nodes. Each node performs the steps of the algorithm based on local information. Thus the algorithm is distributed. Some exchanges of messages are required in this algorithm. Once we have the completion of the algorithm the network is organized into clusters and each node ends up assuming one of these roles; it may remain an ordinary node (guards), it may become a gateway node (centers), it may become a substitute (quarter substitute or center substitute), or it may become the cluster head (quarter) node position. The significance of these roles will become apparent as the algorithm is described.

We assume that each node in the network uses a Global Positioning System or has the position information available and each node is identified by a unique integer from 1 to N and is allowed to transmit control messages related to the algorithm.

During execution of the tag clustering algorithm each node maintains the following

data structures: two lists called, respectively, SINGLENODE, NEIGHNODES, two matrix called CONNECTIVITY and DISTANCES, a variable POSITION, and an indicator named NODESTATUS. These data structures are updated routinely as control messages are received from other nodes. We first describe what these data structures represent.

The CONNECTIVITY matrix has binary entries. A value of 1 in the (i, j) th position indicates the existence of a link between nodes i and j , and a value of 0 the lack of one. Thus, the i th row of the matrix indicates the connectivities of the i th node. It is possible that at the location of node i several rows of the matrix remain unfilled due to the lack of knowledge about the connectivities at distant locations in the network. The variable POSITION gives us the position in the coordinates (h, k) of the node and we can calculate the distances between the nodes using:

$$D = \sqrt{(i - h)^2 + (j - k)^2}. \quad (3.1)$$

After this, we have the DISTANCES matrix whose entries are obtained by (3.1). Thus, the (i, j) th element of the matrix indicates the distances between the i th node and the j th node.

The variable QUARTER designates the identity of the node that is assigned to be the clusterhead for a given node. The NODESTATUS indicator takes on one of four values, GUARD, CENTER, SUBSTITUTECENTER, SUBSTITUTEQUARTER or QUARTER to specify the role assumed by a given node. The NEIGHNODES list includes all neighboring nodes that a node can hear. The SINGLENODE list records those nodes that are not connected to a any node.

In the beginning the algorithm proceeds as follows.

Nodes broadcasts their position and compute the distances between them.

Node i broadcasts the identities of the nodes it has heard from 1 hop, during the earlier broadcast (POSITION and DISTANCES). Thus, node i also receives partial connectivity information of the nodes that it can hear. So at the end of this frame, node i has filled in some of the entries of its connectivity matrix. In particular, it can fill in elements (i, j) of the i th row that satisfy $j > i$. The element (i, j) is set equal to 1 if node i hears from node j and node i appears in the NEIGHNODES list broadcast by node j .

Node i can perform the first stage of the tag clustering algorithm to form clusters and the second stage to determine the NODESTATUS, using its row connectivity information. We see in the Figure 3.5 and 3.6 the two stages of the tag clustering algorithm. We use the rule that the node with the maximum number of NEIGHNODES among a group of nodes is the first candidate to claim QUARTER status (clusterhead). Thus, node i first checks its connectivity row. If there is two or more nodes with the maximum NEIGHNODES, we use the rule that the node with the lowest identity number among a group of nodes is the first candidate to claim QUARTER status. The other nodes will be substitutes of the cluster head (SUBSTITUTEQUARTER)

After this, we form the cluster with the NEIGHNODES of the clusterhead. Now we start tagging the nodes in each cluster using mathematical theory of sets like intersection of sets. We compute the intersection of all the clusters and with these we have the possible gateways; with the matrix DISTANCES we know the distance between the clusterhead of the different clusters and the possible gateways, we use the rule that the node of the set of possible gateways with the lowest distance to the clusterhead is the candidate to claim the CENTER status. The other nodes of the set of possible gateways will be substitute of the CENTER (SUBSTITUTECENTER). The following nodes will be GUARD. At the end we have the complete tagging of the nodes within the cluster.

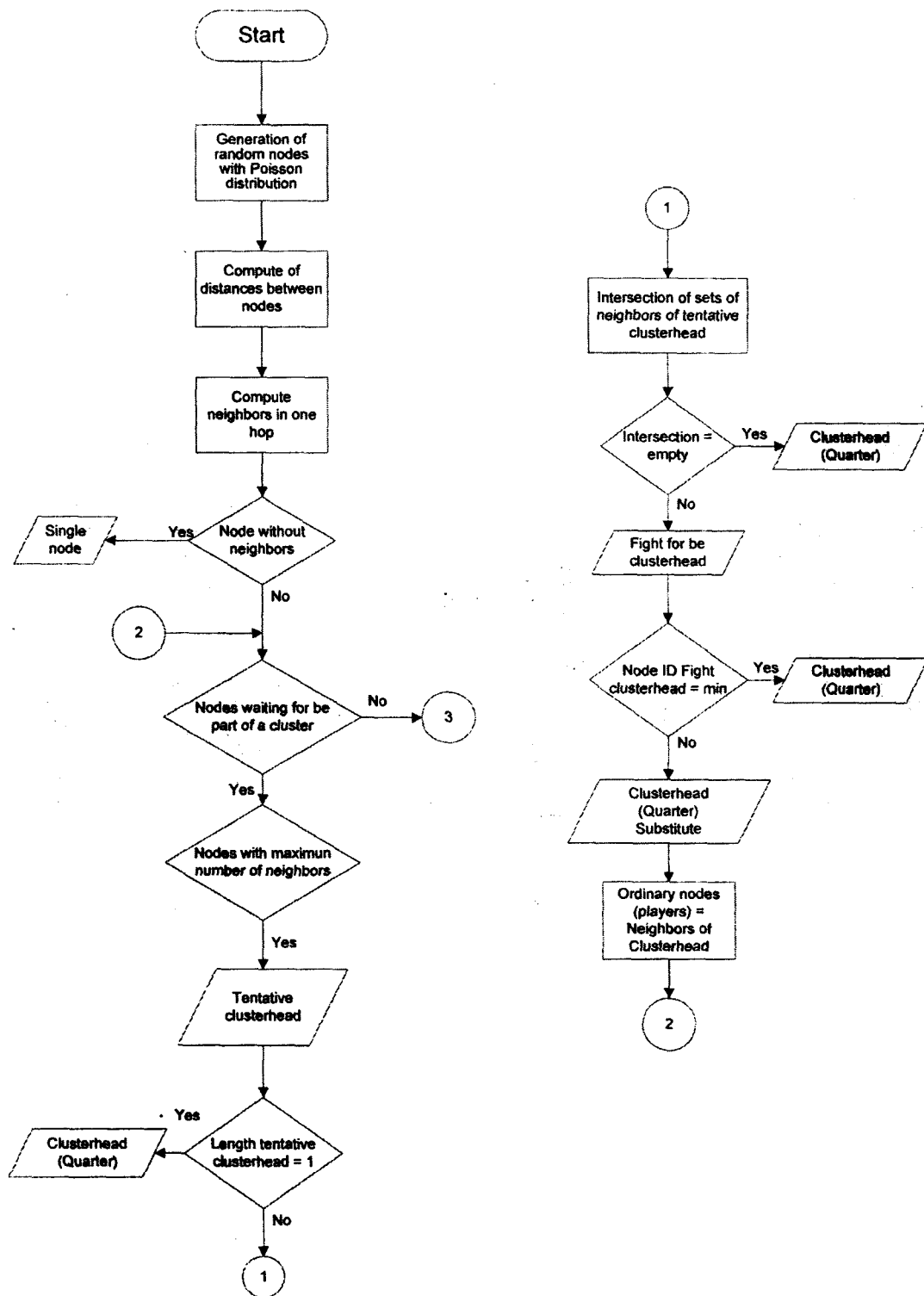


Figure 3.5: Formation of clusters in tag clustering algorithm

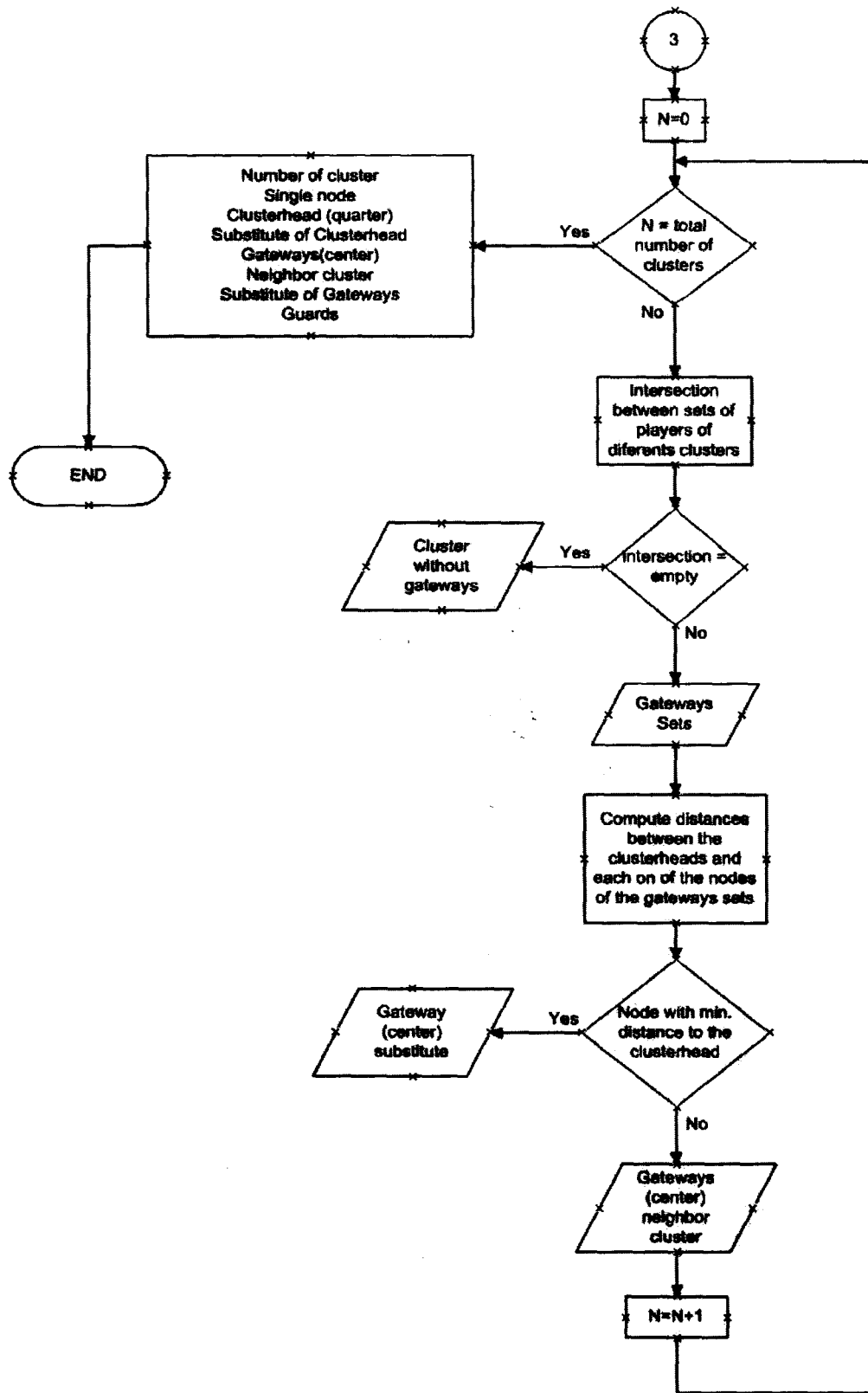


Figure 3.6: Tagging nodes in tag clustering algorithm

Chapter 4

Performance Evaluation

A simulator model was constructed to provide examples of ad hoc networks using the tag clustering algorithm just described, and to depict the resulting organization. In this chapter, we show selected samples of simulation outputs that illustrate several of the possibilities that may arise and the corresponding action by the algorithm. The input parameters for the simulator are the number of nodes, the communication range, a seed number for generating random node positions and the maximum spatial size of the network.

We generate a random network whose nodes are randomly distributed according to a spatial Poisson point process of density λ users per square meters. We simulate a system of n nodes on a 1000 x 1000 square meters. The nodes could move in all possible directions with speed varying uniformly between 0 to a maximum value (m/s). In our simulation experiments, n was varied between 50 and 200, and the nodes moved randomly in all possible directions.

We compare the performance of tag clustering algorithm with seven performance metrics: (i) the number of clusterheads, (ii) the number of Single nodes, (iii) the number of gateways, (iv) the number of ordinary nodes, (v) the number of clusterhead substitutes, (vi) the number of gateways substitutes, and (vii) reachable clusters. The number of clusterheads in the network defines the dominant set. These parameters are studied for varying number of nodes (n) in the system and maximum displacement.

In the first simulation (it is our most important simulation) the nodes were then allowed to move at random in any direction at a speed no greater than 10 m/s. The simulation ran for 3600 seconds, and the network was sampled every second. At each sample time the proposed Tag Clustering Algorithm was run to determine clusterheads and there associated nodes (ordinary nodes, gateways, clusterhead substitute and gateway substitute). For every simulation run the performance metrics mentioned before were measured for the entire 3600 second of simulation. Some of the more noteworthy simulation performance metrics measured are shown in Table 4.1. These performance

metrics provides a basis for evaluating the performance of the proposed algorithm.

Table 4.1: Static Results for 3600 seconds

Average	Communication radio of 150m		
	Maximum	Minimum	Mean
Single nodes	5	0	0.9172
Clusterhead	25	12	18.6722
Gateways	58	19	37.3689
Clusterhead substitute	13	0	3.5347
Gateways substitute	30	3	13.9661
Ordinary nodes	52	7	25.5408
Reachable clusters	3.4375	1.25	2.3096
Nodes per cluster	8.1667	4	5.3562
Gateways per cluster	2.8421	1.1875	2.0056
Guards per cluster	4.3333	0.3182	1.4006
Gateways substitute per cluster	1.8125	0.1429	0.7579
Clusterhead substitute per cluster	0.7059	0	0.1922

The objectives when designing the clustering algorithm were the following: The algorithm must be distributed, since every node in the network only has local knowledge. The algorithm should scale well as the size of the network increases. The created clusters should be reasonably efficient, that is, the selected clusterheads should cover a large number of nodes. If the clustering structure becomes too complex (too many clusters), the number of messages needed to maintain the routing structure would cause congestion in the network.

In the following sections we compare the algorithm in different situations varying the number of nodes, the communication range and the speed of the mobility.

4.1 Proposed Scenarios

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

- Scenario 1, the objective of this scenario is to compare the performance metrics when the total number of nodes and the communications range in the network change. It is important to mention that the nodes in this scenario have no mobility.

- Scenario 2, in this scenario, we compare the cluster performance measures, when we vary the velocity of the nodes, the total number of nodes and the communications range.

4.1.1 Scenario 1

This scenario consists of static simulations of the algorithm, it is important to say that in every slot of time, we generate random nodes with a spatial Poisson distribution, with no mobility; the objective is to observe the behavior to drastic changes of the topology of the ad-hoc network. The simulations are made in 50 slots of time, at each sample time the proposed Tag Clustering Algorithm was run to determine clusterheads and there associated nodes (ordinary nodes, gateways, clusterhead substitute and gateway substitute). Varying the number of nodes in the system, we tried to cover low density, medium density and high density networks. The low density is conformed by an approximated number of 50 nodes, we deduced that we will find a considerable number of single nodes and a low number of clusters, with this the algorithm will be exposed to low communication. The medium density is conformed by an approximated number of 100 nodes, this node density is considered stable in the communication and with a low number of single nodes. We chose to vary the communication range. At the beginning the range is of 150 m and its behavior is observed, then we change the range to 200 m and we observe the changes in the performance parameters. The high density is conformed by an approximated number of 200 nodes, where we see that the number of single nodes will be almost null and the number of clusters will be increased, with this we want to observe the behavior of the tag clustering algorithm in high density and check the growth of the number of clusters.

Table 4.2 shows the parameters of performance for the 3 types of densities. We observed that the number of single nodes is greater in the low density and is almost null for high density. The number of clusterheads is equal to the number of clusters since we only have one clusterhead per cluster, we observe that an approximated increase of 30 percent of the number of clusters when we increase the density and this indicates a stability of the algorithm because the number of clusters is not increased considerably when we have high density. The number of gateways indicates the possibilities of connection with others clusters, since it had been predicted in low density we do not have a considerable connection, and in the high density the connection grows considerably, allowing to find routes between clusters. The substitutes indicate the possibilities of saving routes in case of the death of some of the clusterhead or gateway, the most important point is the increase in the number of gateway substitutes in high density this provides communications to almost all the nodes in the system. In order to determine certain form of robustness in the algorithm, we propose a performance parameter

based on the average of reachable clusters for this we considers the average of reachable clusters per clusters and the average of this in the 50 time slot, it is observed that the number of reachable clusters for different densities increases when we increase the density.

Table 4.2: Static Results for different node density

Average	Communication range of 150m		
	50 nodes	100 nodes	200 nodes
Single nodes	3.08	0.76	0.04
Clusterhead	15.6	20.64	26
Gateways	10.36	29.36	72.22
Clusterhead substitute	3.48	4.46	5.22
Gateways substitute	0.82	5.52	43.8
Ordinary nodes	15.72	29.86	54.08
Reachable clusters	1.228528653	1.940592021	3.518796012

Figure 4.1 shows the CDF of single nodes for the 3 user densities, we observed a greater number of jumps in the distribution for low density that indicates a greater number of single nodes, with 9 single nodes in the worse case. For medium density, we have 4 single nodes in the worst case, and for high density, we have 1 single node. Figure 4.2 shows the probability of having single nodes and a average of clusterhead/ordinary nodes in the system, it is easy to observe the decay of the probabilities as we increase the density.

We can observe the diminution in the average of clusterhead/ordinary nodes if we increase the density, in Figure 4.1 since the number of ordinary nodes almost increases to the double when the density grows, but the number of clusterhead is increased in a 30 percent. On the other hand we have Figure 4.3 that shows the possibility of rescuing a gatewaey in case that it dies, with this factor has trustworthiness in the different connections between clusters. We observed the increase when we increment the densities.

Figure 4.4 shows the robustness of the algorithm for different densities, we concluded that a robustness of 2 is acceptable for tag clustering static, because in high density the robustness is near 4 and with medium density increasing the radius of communication to 200m is bigger than 2 reaching an acceptable communication between clusters. We was simulated with the same communication range (150m) and we found logical results. We can not get a good conclusion only with static simulation, we have to simulated the network in a mobility environment, in the next section we will check this.

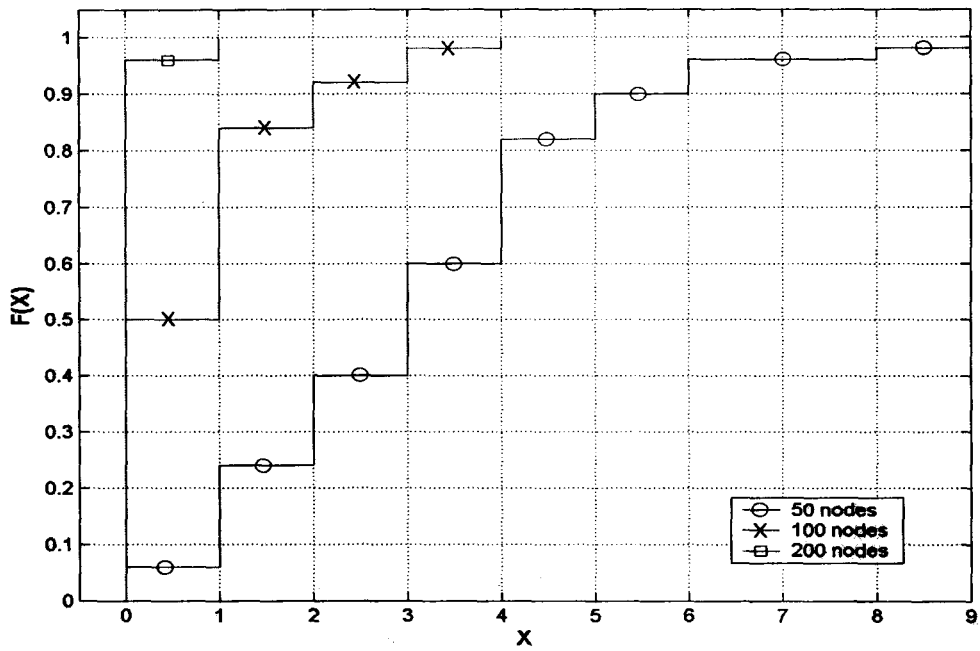


Figure 4.1: Empirical CDF for Single Nodes

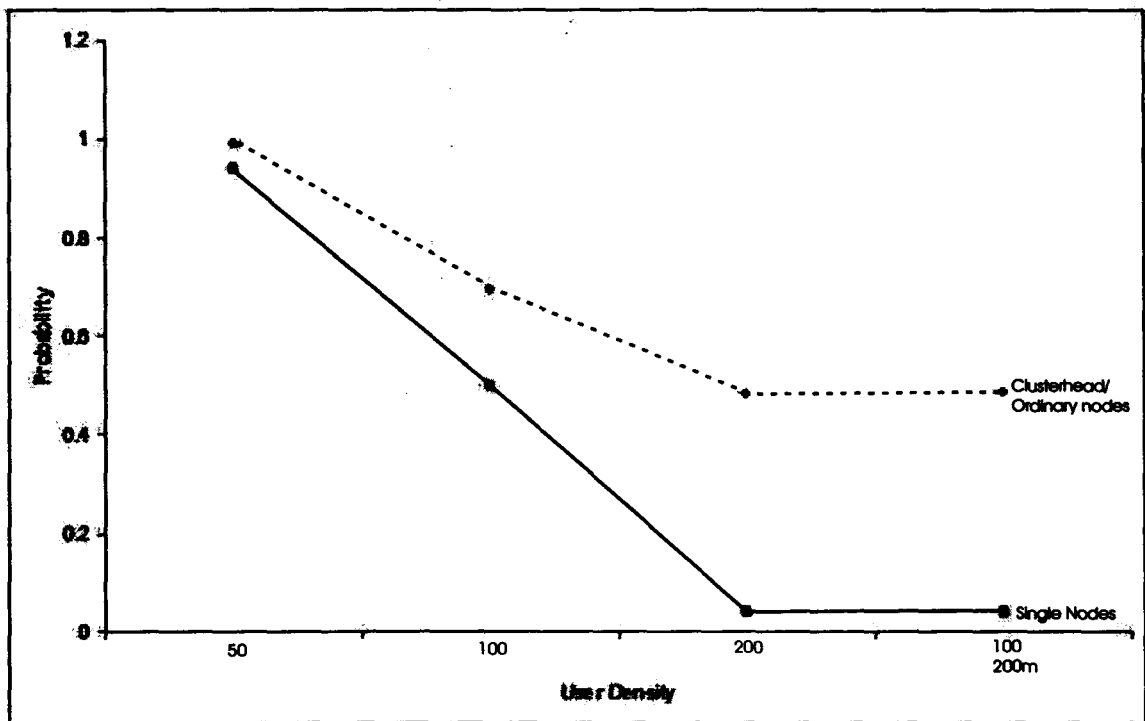


Figure 4.2: Connectivity Weakness Probability

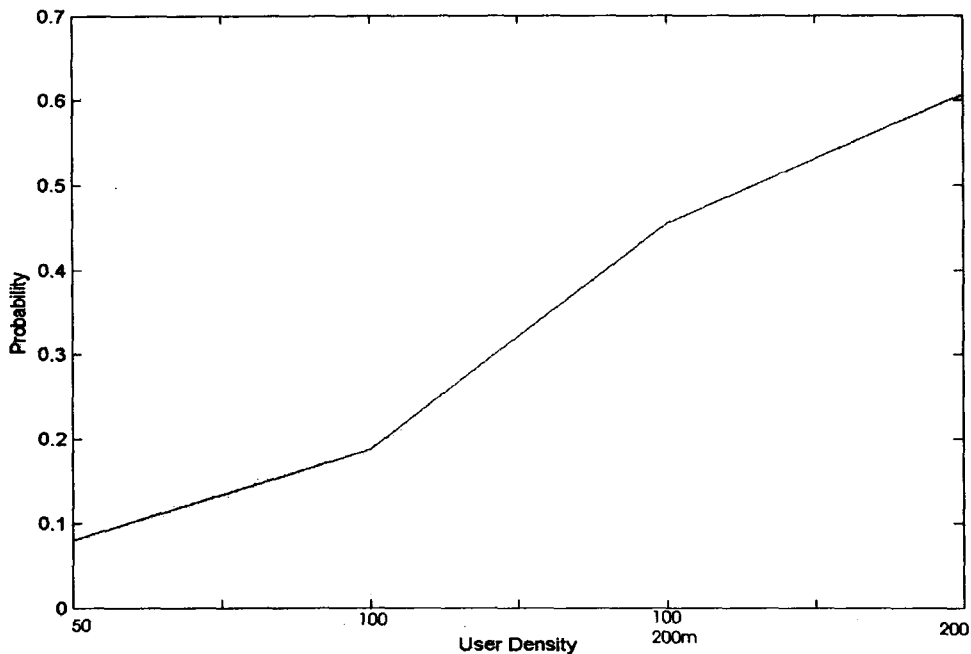


Figure 4.3: Connectivity Probability

We found an acceptable behavior of the algorithm for scenario 1, now we will observe the behavior when we varied the radius of communication to 200 m for the medium density, in the Table 4.3 we observed the performance metrics.

Figures 4.2 to 4.4 also show the behavior for the communication range of 200 m. We can observe a similar behavior between high density and the communication range of 200 m. The communication range will be a control variable to increase the robustness when it will be necessary. In the other metrics we have almost the same behavior.

The service provider could change this control variable, sending a message to the quarter to increase the communication range, if the number of robustness is less than 2. In a real ad-hoc network environment will have other variables that we do not take into consideration because it will be part of a further work.

4.1.2 Scenario 2

This scenario consists of simulations of the algorithm in movement, in time slot zero we generated random nodes with spatial Poisson distribution, and in the following time slots nodes move uniformly at speed of 3 to 10 m/s; the objective is to observe the

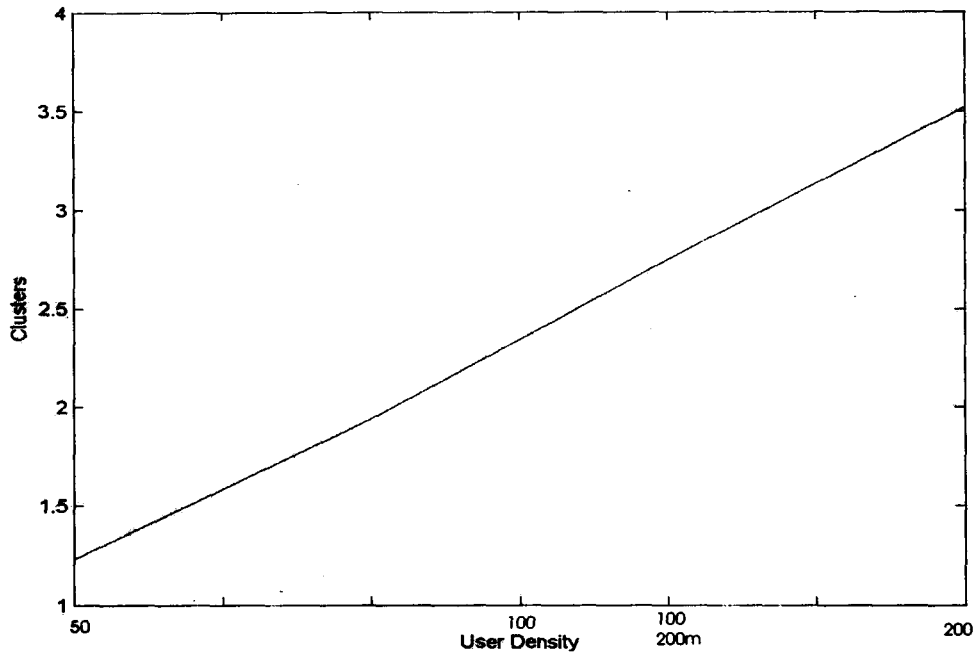


Figure 4.4: Reachable clusters

Table 4.3: Static Results for different communication range

Average	Communication range of 150m and 200m	
	100 nodes (150m)	100 nodes (200m)
Single nodes	0.76	0.04
Clusterhead	20.64	14.02
Gateways	29.36	30.68
Clusterhead substitute	4.46	3.22
Gateways substitute	5.52	13.96
Ordinary nodes	29.86	28.68
Reachable clusters	1.940592021	2.750901574

behavior of the topology of the network when we have movement. The simulations are made in 50 slots of time, at each sample time the proposed Tag Clustering Algorithm was run to determine clusterheads and there associated nodes. We will use the same considerations of densities that we took for scenario 1, the difference will be that we will have mobility with variations of speed from 3 to 10 m/s).

In the Table 4.4 we have the results obtained for the different densities, we observed similar behavior to those of scene 1 in the parameters of single nodes, clusterhead, gateways, gateways substitute and ordinary nodes. Figure 4.5 shows to the CDF for single nodes showing again that with low density obtains the greater number of single nodes, with 5 for the worst cases. For medium density we have 2 as worst case and for high density we do not have single nodes.

Table 4.4: Movement Results for different node density (speed 3 to 10 m/s)

Average	Communication range of 150m		
	50 nodes	100 nodes	200 nodes
Single nodes	2.58	0.7	0
Clusterhead	12.8	18.86	22.7
Gateways	12.72	36.98	70
Clusterhead substitute	2.8	2.66	4.66
Gateways substitute	2.24	14.88	60.84
Ordinary nodes	16.86	25.92	41.8
Reachable clusters	1.210149374	2.345787785	3.471350029

Figure 4.6 shows the probability of existence of single nodes, where we found a decay again as we increased the densities; in these graphs we can observe different additional parameters from the 3 densities, among them we have 100 nodes in 3600 seconds of simulation, with different speed and different radius from cover. The first parameter will give us the behavior in large slot of time and show that the algorithm is stable for all the times; the second parameter will give us the stability of the algorithm for high speeds (23 to 30 m/s), but is know that a ad hoc network will not work in high speed. Ahead we will discuss the results obtained for these new parameters.

We can observe the diminution the average of clusterhead/ordinary nodes if we increased the density, in Figure 4.6 we observed since the number of ordinary nodes almost increases to the double when the density grows, but the number of clusterhead is increased less than 30 percent, that is why we have closest values. On the other hand we have Figure 4.7 that shows the possibility of rescuing a gateway in case that it dies,

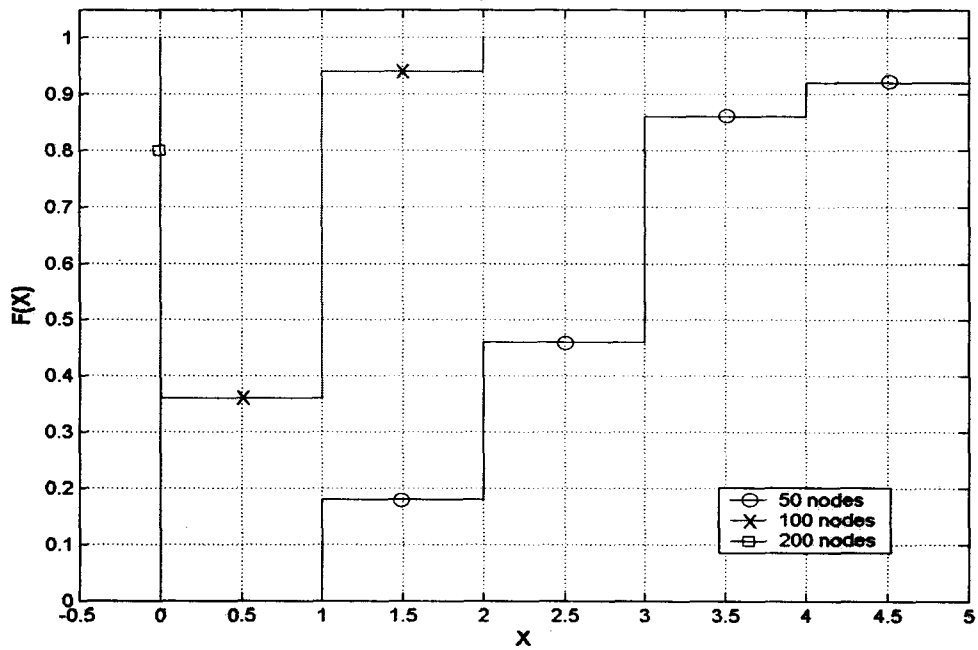


Figure 4.5: Empirical CDF for Single Nodes

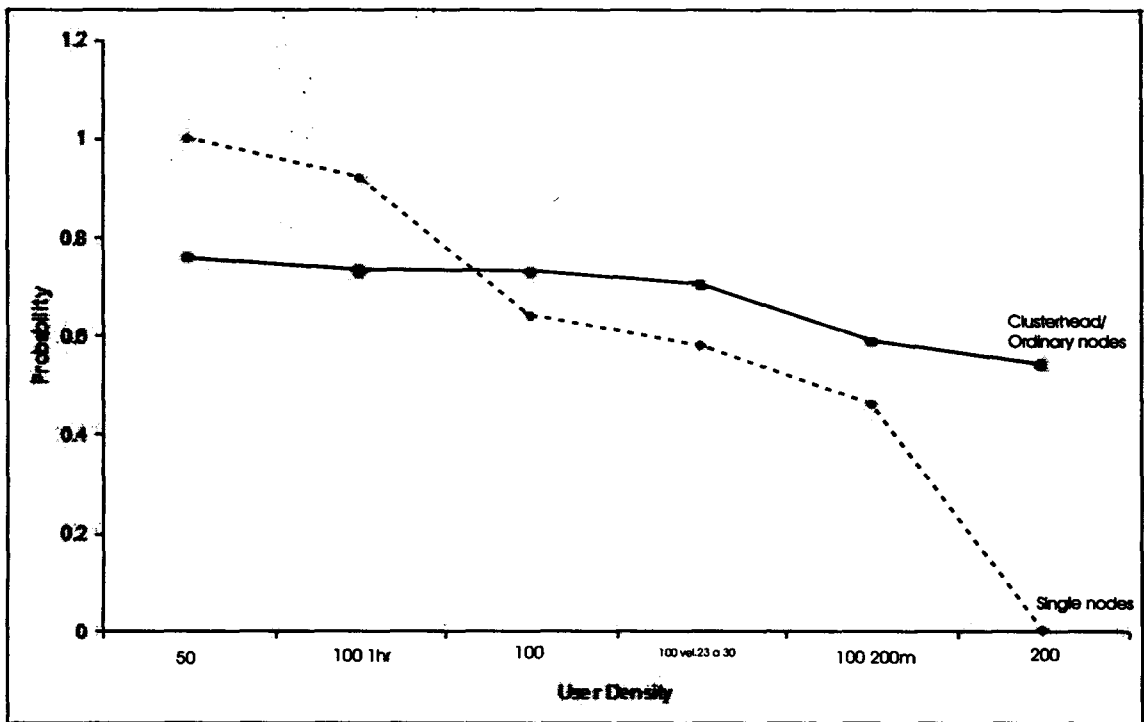


Figure 4.6: Connectivity Weakness Probability

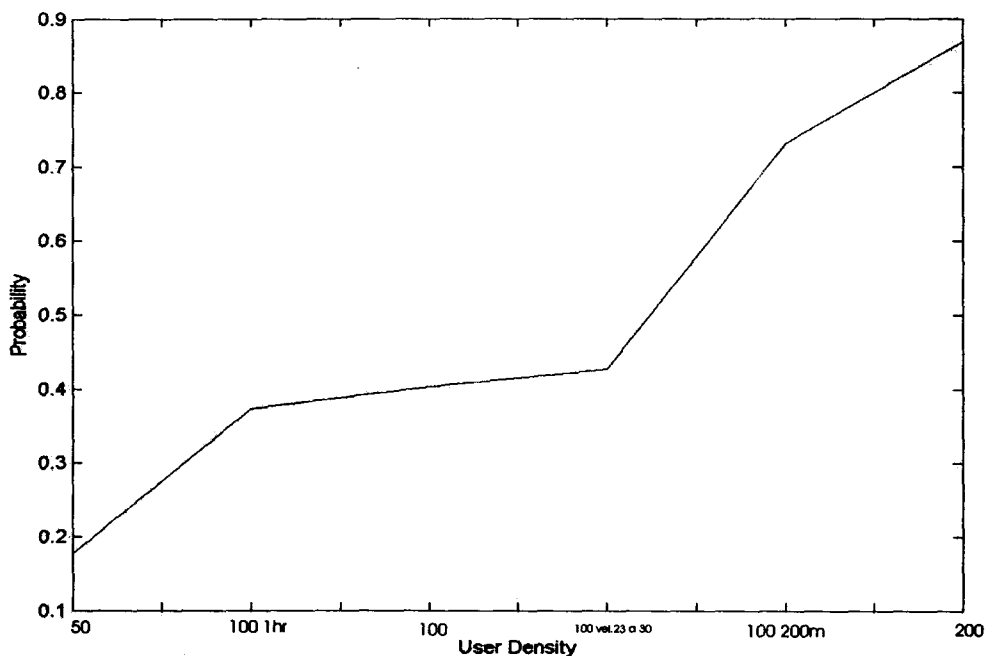


Figure 4.7: Connectivity Probability

with this factor has trustworthiness in the different connections between clusters. We observed the increase when we increment the densities.

Figure 4.8 shows the robustness to us of the algorithm for different densities, we concluded that a robustness of 3 is acceptable for tag clustering in movement, because in high density the robustness is greater to 3,5 and with medium density increasing the radius of communication is greater to 3 reaching an acceptable communication between clusters.

We observe closest values between scenario 1 and 2, and the same behavior in the figures, this will help us to get a good conclusion of the performance of the tag clustering algorithm.

Table 4.5 and Figures 4.6 to 4.8 also show the behavior for the 3600 seconds simulation, speed of 23 to 30 m/s and communication range of 200 m. We can observe a similar behavior between them. The communication range will be a control variable to increase the robustness when it will be necessary. The service provider could change this control variable, sending a message to the clusterhead to increase the communication range, if the number of robustness is less than 3. In a real ad-hoc network environment

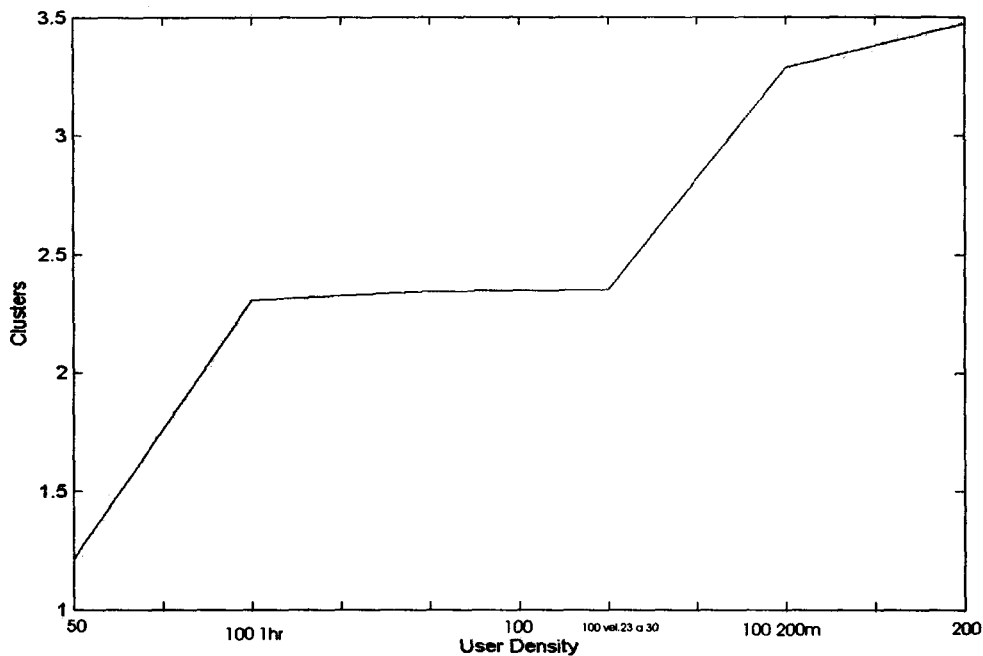


Figure 4.8: Reachable clusters

will be have other variables that we do not take to consideration because it will be part of a further work.

We found an acceptable behavior of the algorithm for scenario 2 now we can use tag clustering for routing packets between the nodes, now we will observe the behavior when we find the route between two nodes using dijkstra algorithm and a approximation of the tag routing algorithm. We will proposed the general idea for the routing in tag clustering, the performance of this will be part of further work.

4.2 Finding the Shortest Path

The general idea for the tag routing algorithm is that the nodes that belongs a cluster has the routing table of the cluster. The clusterhead will have the statistic table and substitute table. The gateway will have the cluster neighbor table.

If a node want to send a packet to another node, first they have to send it to the clusterhead, the clusterhead taking account the statistic will decide the best route; if

Table 4.5: Movement Results for different node density and speed

Average	Communication range of 150m		
	100 nodes(3-10 m/s)	100 nodes(23-30 m/s)	100 nodes(200 m)
Single nodes	0.7	1.18	0.46
Clusterhead	18.86	17.98	12.86
Gateway	36.98	36.52	36.24
Clusterhead substitute	2.66	3.2	2.12
Gateway substitute	14.88	15.6	26.5
Ordinary nodes	25.92	25.52	21.82
Reachable clusters	2.345787785	2.351151162	3.287317493

the destination node belongs to the cluster the clusterhead will send the packet, if not it will send the information to the gateway. The gateway will check in there cluster neighbor table the gateway of the other clusters, and it will send it to them. Now the gateway will send the the packet to the clusterhead and will use the same idea described before.

In Figure 4.9 show the shortest path using dijkstra algorithm for a 100 node density and 200 m of communication range. Figure 4.10 show the shortest path using dijkstra algorithm with the same node density and network topology. With a communication range of 150 m. Figure 4.11 show tag routing in the same node density and network topology using tag clustering algorithm. With a communication range of 150m. In all the cases node 92 wants to send a data packets to node 5.

4.3 Communication Overhead

Tag clustering algorithm is a hybrid algorithm that use concepts of the Lowest-ID and the Highest-connectivity clustering algorithms, for this the communication overhead will be the same that them. Node mobility can cause topology changes (link/cluster additions/deletions) to propagate up to any level. Despite this complication, John Sucec and Ivan Marsic [15] derived the following break-downs of the control overhead. The values are expressed as packet transmissions per node per second unless otherwise specified.

- $\phi_{\text{Hello}} = \theta(1)$ - "Hello" protocol
- $\phi_{\text{CL}} = O(\log(N))$ - cluster formation and maintenance messaging.

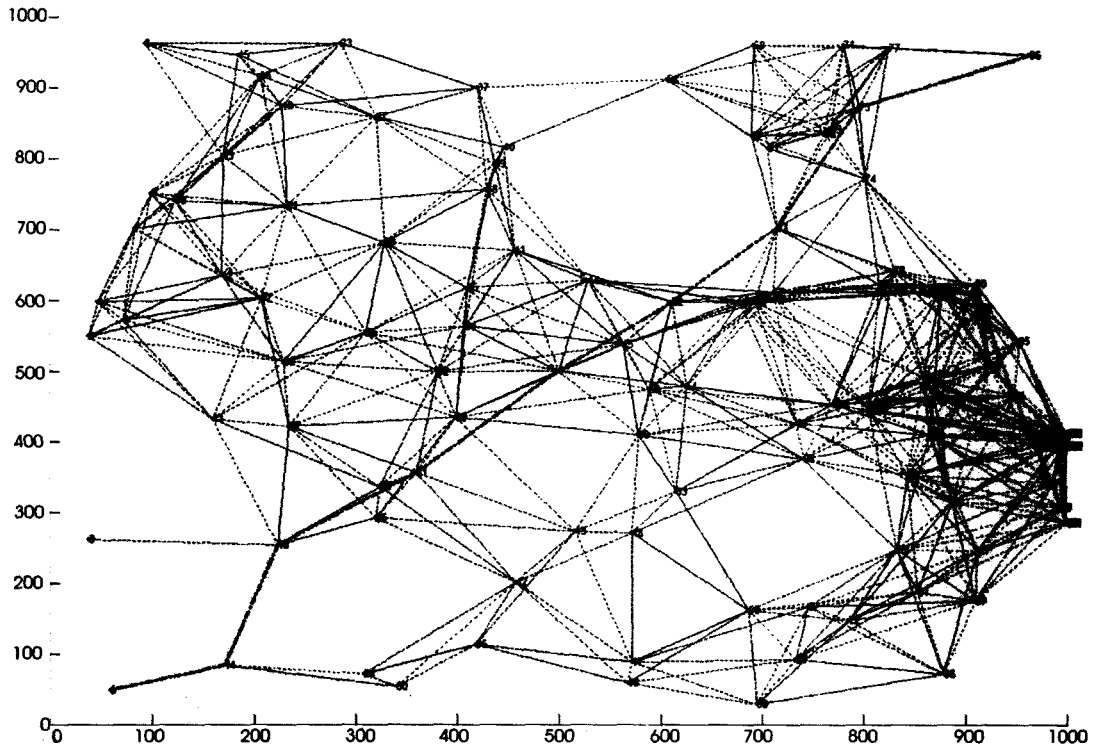


Figure 4.9: Shortest Path communication range 200m without clustering

- $\phi_{ACQ} = O(\log(N))$ - acquisition of local data when node migrates from one cluster to another.
- $\phi_{Flood} = O(\log(N))$ - flooding of cluster topology updates to cluster members.
- $\phi_{Reg} = (N)$ - location registration events.
- $\phi_{Handoff} = \theta \log^2(N)$ - handoff or transfer of location management data.
- $\phi_{Qry} = \theta(h)$ - query events.
- $\phi_{Ctrl-Header} = \theta \log(N)$ - bits per control message datagram - addressing information required in datagram headers.

In summary, the total communication overhead in such a hierarchical routing protocol is $\theta \log^2(N)$ packet transmissions per node per second and $\theta \log^3(N)$ bits per node per second, assuming that the node density, mobility, and traffic load remain constant. This is a substantial (exponential) improvement over the linear costs of routing in ad hoc networks. We propose in the chapter 5 an analysis of the overhead produced by the statistics.

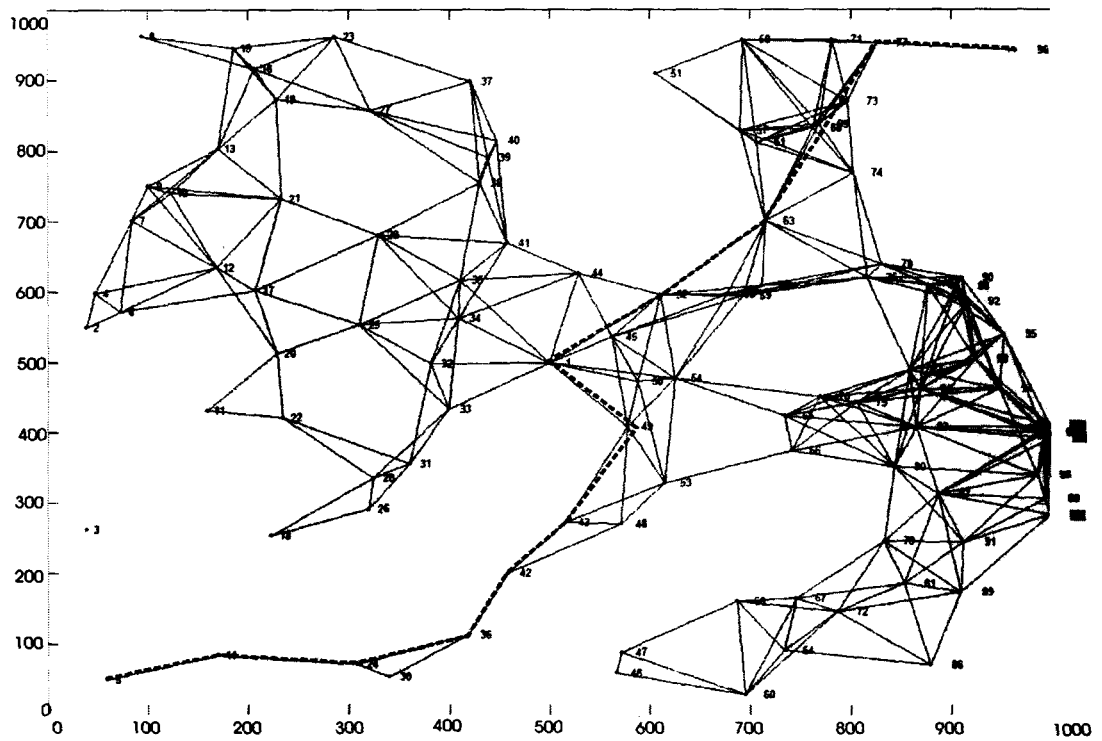


Figure 4.10: Shortest Path communication range 150m without clustering

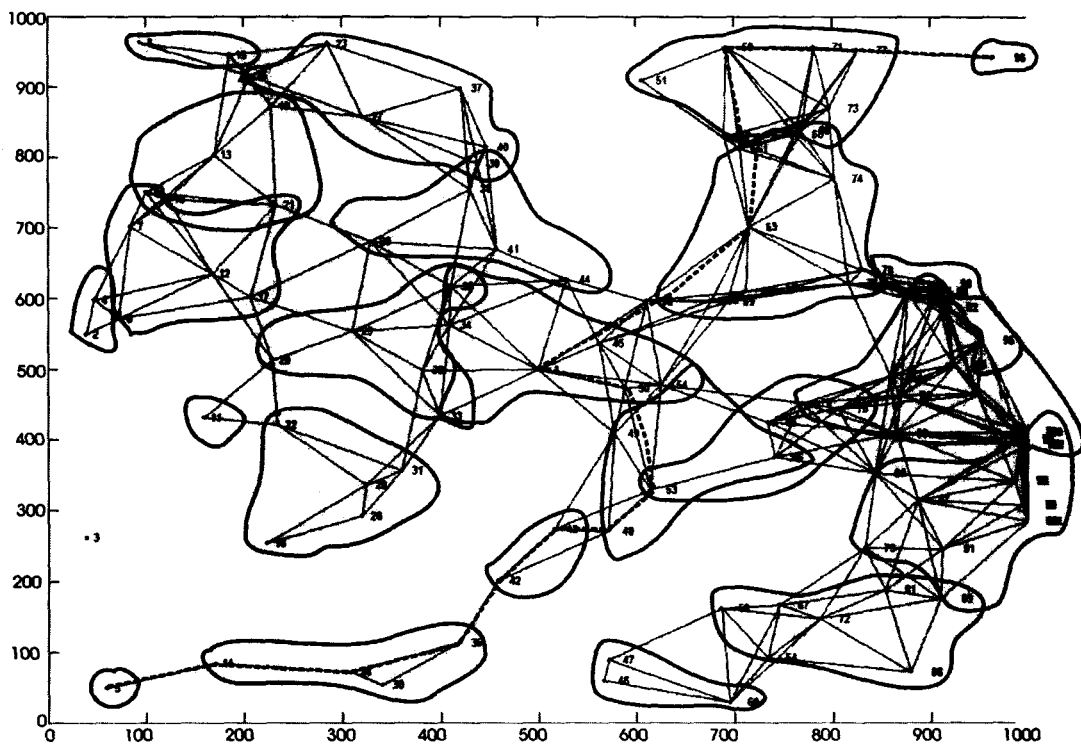


Figure 4.11: Shortest Path communication range 150m with tag clustering

4.4 Tag Clustering Algorithm vs Different Clustering Algorithms.

We cannot have a comparison between tag clustering and different clustering algorithms because the main idea of tag clustering is the cooperative work, as we explain in the chapter 2, that is a new topic in the related to topology generation, organization and maintenance. For this in Table 4.6 we show in a general manner the different clustering algorithms [8].

Table 4.6: Summary of Different Clustering Algorithms

Algorithm	Properties	Complexity	Strengths	Weaknesses
Lowest-ID (LCA2)	Clusterhead selection based on node ID. Clusterhead is directly linked to any other node in the cluster.	Constante time complexity, message complexity increase with denseness of graphs	Fast and simple algorithm. Relatively stable clusters.	Small clusters. Some clusterheads likely to ramian for long time.
Highest-connectivity	Clusterhead selection based on highest degree, otherwise same as LCA2	Same as LCA2	The nodes with highest degree are good candidates for clusterheads	Very unstable clusters.
Tag clustering	Clusterhead selection based on highest degree, if we have nodes with the same highest degree the node with the min ID will be the CH.	Same as LCA2	Cooperative work. Taggin of the nodes.	Further work
Max-min D-cluster	Cluster radius d , where d is a constant.	$O(d)$ time and clusters	Large and stable	High number of messages sent.
Hierarchical	No fixed diameter of each cluster. Cluster size $< k, 2k - 1 >$, where k is a constant.	Time complexity $O(E)$	Guaranteed upper and lower bound on cluster size.	Slow algorithm. Cluster radius can be up to k . algorithm
Adaptive clusters	Created clusters should be connected in time t with probability α	Undefined.	α and t can be varied in order to adapt to different mobility rates.	Difficult to predict future connectivity.

Chapter 5

Conclusion and Further Research

This thesis presents in section 5.1 the conclusions of the thesis, and section 5.2 some future research.

5.1 Conclusions

In the present work we design a new clustering algorithm for ad hoc networks called Tag clustering algorithm (based in cooperative work), like an efficient alternative to the clustering algorithms commonly used. Diverse scenarios were analyzed varying the density of the network, the communication range and finally the mobility of the nodes. By means of the analysis of the thrown results of the simulations for the different scenarios, the following conclusions are obtained:

- Number of reachable neighbors. It is observed an increase in its number as is increased the density of the network and mobility. We concluded that to robustness of 2 is acceptable for tag clustering static, and to robustness of 3 is acceptable for tag clustering in movement.
- Number of clusters. The objective of the tag clustering algorithm is to find a feasible interconnected set of clusters covering the entire node population. The algorithm scale well as the size of the network increases. We observed that an approximated increase of 30 percent of the number of cluster when we increased the density and this indicates a stability of the algorithm because the number of clusters is not increased considerably when we have high density. With this we avoid that clustering structure becomes too complex (too many clusters), and the number of messages needed to maintain the routing structure would not cause congestion in the network.
- Trustworthiness of the connection. With the substitutes indicates the possibilities of saving routes in case of the death of some of the clusterhead or gateway, the most important point is the increase in the number of gateway substitutes in high density this provides communications to almost all the nodes in the system.

- Single nodes. Single nodes is easy to observe the decay of the probabilities as we increased the density. If we want avoid single node, with the control variable (communication range) we can decrease the number of single nodes.
- Routes. We can find routes in the tag clustering algorithm using the general idea for the tag routing using cooperative work.
- QoS. For medium density and speed of 3 to 10 m/s we have a good performance in the conectivity and low conectivity weakness. And with the control variable (communication range) we can improve this parameters increasing the communication range.
- Processing time. When we ran 3600 seconds of simulation with tag clustering, our processing time was the same 3600 seconds it is 1 second of processing with tag clustering, but this second include movement of the nodes, random location and tag clustering, so we considered that the processing time of tag clustering is below of 50 ms.
- Types of Ad Hoc Networks. We can use tag clustering in networks with good CPU and battery life, like packet radio networks, personal communication systems, mesh networks and wireless local are networks, because we need small processing time for the topology generation, organization and maintenance. Today in sensor networks we can not use tag clustering, for the small CPU; maybe in the future when we increse the CPU of the sensor we could use tag clustering.

5.2 Further Research

Once the organization described above is achieved via cooperative work in tag clustering algorithm, there are several issues that arise relative to the operation of the network. In this section we offer a preliminary discussion of some of them.

- Security, Security is an important issue for ad hoc networks, especially for those security-sensitive applications. To secure an ad hoc network, we consider the following attributes: availability (ensures the survivability of network services despite denial of service attacks), confidentiality (ensures that certain information is never disclosed to unauthorized entities), integrity (guarantees that a message being transferred is never corrupted), authentication (enables a node to ensure the identity of the peer node it is communicating with), and non-repudiation (ensures that the origin of a message cannot deny having sent the message). For this we can introduced the concept of referees in order to maintain the integrity and security of clusters. The attributions and responsibilities of the referee will

be: If a node wants to belong a cluster it must request permission to the referee first and this it will decide with base to the information contained in the control table if the access is allowed or no. To inspect and to approve all the packets that are being sent in the network. Not to allow that intruder node uses routes that can cause a breaking in the internal security of the network. It has the faculty to stop a transmission when the circumstances justify it (reasons for security).

- **Bandwidth, Analysis of the consumption of bandwidth that brings the algorithm. Analysis of the overhead of the statistics interchange and control table.**
- **Statistic Analysis, With this we can make decisions and see the behavior of the the control and observable variables. The control variables can be changes by the clusterhead depending of the performance metrics. In real ad-hoc network environment will be have other variables that we don not take to consideration, we proposed applying this variables to the algorithm using Ns2 simulation.**
- **Analytic model of the performance of the network. We propose Cluster Splitting. Since too small a cluster size would underutilize the available resources while too large a cluster size would increase the storage and communication overhead, a suitable cluster size should be determined in order to achieve a good performance. Splitting larger clusters and merging smaller ones will make cluster size adjustments.**
- **QoS. Analysis of the consumption of battery when we use tag clustering and when we increase the communication range. Analysis per node status of the consumption of battery and CPU. Analysis with distributed work between clusterhead and clusterhead substitute, gateway and gateway substitute.**

Bibliography

- [1] Taieb F. Znati A. Bruce McDonald, "A mobility-based framework for adaptive clustering in wireless ad hoc networks," *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS*, vol.17, No.8, pp. 1466–1487, August 1999.
- [2] J.E. Wieselthier A. Ephremides and D.J. Baker, "A design concept for reliable mobile radio networks with distributed control," *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS*, vol. 2, pp. 226–237, 1984.
- [3] Ephremides A. Baker, D.J. and J.A. Flynn, "The design and simulation of a mobile radio network with distributed control," *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS*, pp. 226–237, 1984.
- [4] Dimitri Bertsekas and Robert Gallager, *Data Networks*, Prentice Hall, 1987.
- [5] Ivan Stojmenovic David Simplot-Ryl, "Ad hoc networking: Data communications and topology control," *IEEE Network*, pp. 4–5, July–August 2004.
- [6] Mieso H. Denko, "The use of mobile agents for clustering in mobile ad hoc networks," *Proceedings of SAICSIT*, pp. 241–247, 2003.
- [7] M. Gerla and J.T. Tsai, "Multicluster, mobile, multimedia radio networks," *ACM/Baltzer Journal of Wireless Networks*, Vol. 1, No. 3, pp. 255–265, 1995.
- [8] Tomas Johansson and Lenka Carr-Motycková, "Bandwidth-constrained clustering in ad hoc networks," *Proceedings of The Third Annual Mediterranean Ad Hoc Networking Workshop on Foundations of Mobile Computing*, pp. 379–385, 2005.
- [9] Per Johansson Magnus Frodigh and Peter Larsson, "Wireless ad hoc networking the art of networking without a network," *Ericsson Review*, No. 4, pp. 248–263, 2000.
- [10] Sajal K. Das Mainak Chatterjee and Damla Turgut, "WCA: A weighted clustering algorithm for mobile ad hoc networks," *Cluster Computing*, vol. 5, pp. 193–204, 2002.

- [11] A.K. Parekh, "Selecting routers in ad-hoc wireless networks," *Proceedings of the SBT/IEEE International Telecommunications Symposium*, 1994.
- [12] Charles E. Perkins, *Ad Hoc Networking*, Addison-Wesley, 2001.
- [13] Amitabh Mishra Satyabrata Chakrabarti, "Qos issues in ad hoc wireless networks," *IEEE Communications Magazine*, pp. 142–148, February 2001.
- [14] J. Sharony, "A mobile radio network architecture with dynamically changing topology using virtual subnets," *ACM/Baltzer Mobile Networks and Applications*, No. 1, pp. 75–86, 1996.
- [15] John Sucec and Ivan Marsic, "Clustering overhead for hierarchical routing in mobile ad hoc networks," *IEEE INFOCOM*, pp 1698–1706, 2002.

Centro de Información-Biblioteca



30002006631246