K-HOPS PASSIVE CLUSTERING IN AD HOC NETWORKS

by

Liang Tang

A thesis submitted to
the Faculty of Graduate Studies and Research
in partial fulfillment of
the requirements for the degree of
Master of Computer Science

Ottawa-Carleton Institute for Computer Science
School of Computer Science
Carleton University
Ottawa, Ontario 2005

0-494-06834-5

Abstract

Ad Hoc Networks are expected to play a significant role in future mobile computing applications. For efficient communication among mobile nodes, one promising method is that ad hoc networks be typically grouped into clusters.

In this paper, we broadly analyze existing clustering mechanisms and classified them into seven categories. Furthermore, we propose a new clustering algorithm named **KHPCA** (K-hops Passive Clustering Algorithm). "**K**" is a pre-defined cluster size parameter that can be designated according to different network scenarios. Moreover, we propose the concept of "**Candidate-Clusterheads**" to alleviate the constraints which occur in most current clustering algorithms, such as the clusterheads becoming the bottleneck in network and causing high overheads and unstable network topology due to excessive re-clustering. **KHPCA** is simple and stable. Through our simulation we can conclude that the **KHPCA** functions better than conventional algorithms such as Lowest-ID in terms of network's stability, robustness and capacity of fault tolerance.

Acknowledgments

Firstly, I sincerely thank my advisor Professor Evangelos Kranakis for his continuous help and invaluable guidance. At the beginning, Dr. Evangelos Kranakis was my teacher of the course named "Data Networking" in the fall term of year 2002. I was inspired by his elaborate course material and arresting teaching techniques. His teaching really widened my perspective in computer networking field and enlightened my ambition to fulfill more achievement in this field. Even though my thesis topic is relatively new-fashioned and access to existing research materials limited, Professor Evangelos Kranakis always provides me with invaluable advice, guidance and correction. Honestly, I could not have accomplished my thesis without his warm-hearted help.

I would also like to thank Taek Jin Kwon, the author of "On-demand routing in large ad hoc wireless networks with passive clustering", for providing me valuable internet resources when I dedicated myself to this thesis topic at very beginning time.

Furthermore, I would present million appreciations to my dear parents, grandparents and my darling wife Linda Lee. Your support, love and encouragement always give me endless power and confidence.

Table of Contents

List of figures

Figure **Page**

List of abbreviations

2HPCA	KHPCA with K = 2
3HPCA	KHPCA with K = 3
ABR	Associativity-Based Routing
AODV	Ad Hoc On-Demand Distance-Vector
AOW	Automatic On-demand Weighted algorithm
AP	Access Point
BSS	Base Station Set
CGSR	Clusterhead Gateway Switch Routing
CSL	Cluster Sub Layer
DARPA	Defense Advanced Research Projects Agency
DCA	Distributed Clustering Algorithm
DG	Distributed Gateway
DMAC	Distributed Mobility-Adaptive Clustering Algorithm
DSDV	Destination-Sequenced Distance-Vector
DSR	Dynamic Source Routing
FDW	First Declaration Wins
GloMoSim	Global Mobile Information System Simulator
IEEE	Institute of Electrical and Electronic Engineering
IETF	Internet Engineering Task Force
KHPCA	K-hops Passive Clustering Algorithm
LCC	Least Cluster Change
LOS	Line Of Sight
MAC	Medium Access Control
MANET	Mobile Ad hoc networking
NTDRS	Near Term Digital System
PAN	Personal Area Network
PDA	Personal Digital Assistant
PNNI	Private Network–to–Network Interface
PRnet	Packet Radio Network

Qos	Quality of Service
RERRs	Route Errors
RREPs	Route Replies
RREQs	Route Requests
SSR	Signal Stability Routing
TORA	Temporally Ordered Routing Algorithm
WCA	Weighted Clustering Algorithm
WRP	Wireless Routing Protocol
WLAN	Wireless Local Area Networks
ZRP	Zone Routing Protocol

CHAPTER 1

Introduction

1.1 Background

Wireless communication between mobile users is becoming more and more popular than ever before. This is due to current development of wireless technology and the increasing popularity of portable devices, such as laptops, PDAs (Personal Digital Assistant), and mobile phones.

One major approach for enabling wireless communication is to form an ad hoc network. Ad hoc networking allows mobile devices to establish communication easily and promptly without the aid of a central infrastructure. The roots of ad hoc networking can be traced back to 1968, when work on the ALOHA network was initiated. The goal of the ALOHA network was to connect educational facilities in Hawaii. Although fixed stations were needed and its protocol was a single-hop protocol, that is, it did not support routing and every node had to be within reach of all other participating nodes, the ALOHA protocol lent itself to distributed channel-access management and hence provided a basis for the subsequent development of distributed channel-access schemes that were suitable for ad hoc networking.

Inspired by the ALOHA network, In 1973, DARPA (Defense Advanced Research Projects Agency) began work on the PRnet (Packet Radio Network) [23]. PRnet is a multi-hop network which means that nodes in the network cooperated to relay traffic on behalf of one another to reach distant stations that would otherwise have been out of transmission range. Obviously, the "multi-hoping" techniques increased network capacity and broke the limitation of transmission range of the single-hop model because the spatial domain could be reused for concurrent but physically separate multi-hop sessions. Although many experimental packet-radio networks were later developed, these wireless systems were mainly used in large-scale multi-hop military or rescue operation systems. When developing IEEE 802.11, a standard for wireless local area networks (WLAN), the Institute of Electrical and Electronic Engineering (IEEE) replaced the term Packet-radio network with ad hoc network.

Mobile ad hoc network is a self-organizing network system that contains high dynamic topology structure and randomly moved mobile nodes [41]. We always designate it as "mobile multi-hop network" because there is no direct connection between communication peers (point-to-point or point-to-multipoint) under most circumstances. The successful transmission must rely on the intermediate nodes to relay the messages hop by hop. While the early applications and deployments of ad hoc networking have been military oriented, commercial applications have also grown substantially since then. Besides, the Internet Engineering Task Force (IETF) established a working group named The Mobile Ad hoc networking (MANET) to investigate and develop candidate standard Internet routing support for mobile, wireless IP autonomous segments [55]. The purpose of this working group is to standardize IP routing protocol functionality suitable for wireless routing applications within both static and dynamic topologies.

1.2 Motivation

Due to the rapid progress of mobile communication and mobile terminal technology, ad hoc networking not only has been fully developed in military objectives, but also has been used more and more widely in commercial mobile communications. It is useful for implementing solutions for business, entertainment and safety applications in the office, residential, industrial, and inter-/intra-vehicle communication arenas. The key characteristic of a wireless ad hoc network is the ability of any node to route traffic for other nodes. This is especially useful when those other nodes are portable or mobile. The ad hoc network provides reliable communication in the presence of frequent changes in the network connection topology.

Although ad hoc networking is a new-fashioned technology, it has many unique advantages such as it can be deployed easily and fast and has less dependence on infrastructure. It has to face many significant challenges. The basic set of challenges includes limited wireless transmission range, mobility-induced route changes, security hazards, potentially frequent network partitions, mobile nodes distribution and organization, battery constraints and QoS (Quality of Service).

Therefore, in order to make ad hoc networking an effective and usable tool that can be used more widely, commercially and efficiently, these constraints and challenges must be solved. Consequently, a large body of research has accumulated to address these specific issues and constraints. To present the huge amount of research activities on ad hoc networks in a systematic way, we will use the simplified architecture shown in Figure 1.1.

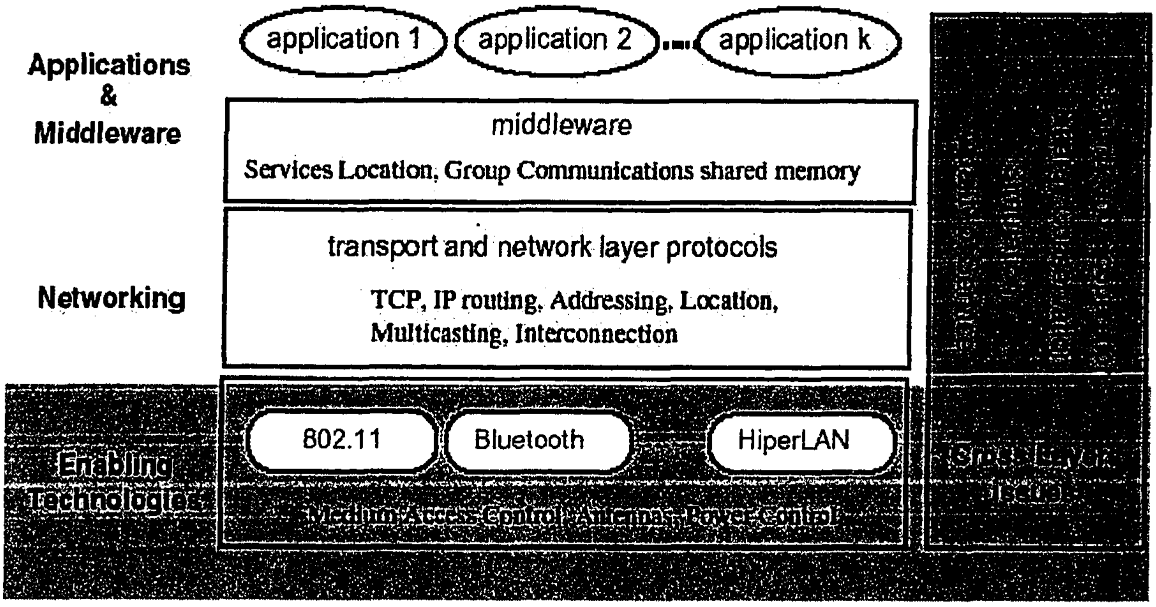


Figure 1.1: A simple architecture of ad hoc network research activities

As shown in the figure above, the research activities can be grouped, according to a layered approach into three main areas:

① Enabling technologies;

② Networking;

③ Middleware and applications.

In addition, as shown in the figure, several issues (energy management, security and cooperation, quality of service, network simulation) span all areas.

This thesis will be dedicated to the second research area, that is, the networking in ad hoc networks. To cope with the self-organizing, dynamic, volatile, peer-to-peer communication environment in an ad hoc network, most of the main functionalities of the Networking protocols (i.e., network and transport protocols in the Internet architecture) need to be re-designed. The aim of the networking protocols is to use the one-hop transmission services provided by the enabling technologies to construct end-to-end delivery services, from a sender to one or more receiver(s). To establish an end-to-end communication, the sender needs to locate the receiver inside the network. Once, a user is located, routing and forwarding algorithms must be provided to route the information through the network. However, the highly dynamic nature of a mobile ad hoc network results in frequent and unpredictable changes of network topology. Hence, this kind of changing adds difficulty and complexity to routing among the mobile nodes. The challenges and complexities, coupled with the critical importance of routing protocols in establishing communications among mobile nodes, make routing area the most active research area within the ad hoc network domain. Numerous routing protocols and algorithms have been proposed, and their performance under various network scenarios and traffic conditions have been studied and compared [45],[48]. Ad hoc networks should support a large number of heterogeneous users since any device with a microprocessor can in principle be an ad hoc network node. In a large network, conventional flat routing mechanisms produce an excessive amount of information that can saturate the network. Furthermore, given the nodes heterogeneity, nodes may have highly variable amount of resources, and this naturally produces a hierarchy in their roles inside the network. Nodes with large computational and communication power, and powerful batteries are more suitable for supporting the ad hoc network functions than small-embedded systems.

Cluster-based routing is a promising solution to address nodes heterogeneity, and to limit the amount of routing information that propagates inside the network. The basic description of clustering mechanism is to group the network nodes into a number of clusters. In a general sense, the clustering problem is one of classifying nodes hierarchically into equivalence classes, according to certain attributes. One goal of clustering could be to reduce route discovery overhead (by address space aggregation or by localizing control messages) to optimize resources like battery power and network capacity, or to simplify addressing and management. Thus, the clustering techniques can be used to construct a hierarchy within an ad hoc network to increase the scalability and management of the network. As a result, a large number of research papers in clustering mechanism have been proposed recently [5],[11],[25],[27],[31],[37],[51][].

1.3 Contribution

This thesis focuses on clustering mechanisms in mobile ad hoc networks. Being interesting and crucial issues in ad hoc networks, clustering techniques have been intensely investigated by many researchers and wireless experts worldwide. Consequently, a large number of researches and clustering algorithms have been developed. Based on the most significant clustering properties, we have investigated and analyzed the dominating clustering mechanisms broadly and thoroughly. Generally, we classified these mechanisms into seven categories which are Adaptive (non-clusterhead based) clustering algorithm, the Lowest-ID clustering algorithm, the Highest-Degree clustering algorithm, the Node-Weight clustering algorithm, the Least Cluster Change clustering algorithm, the Max-Min D-Cluster Formation algorithm, and the passive clustering algorithm respectively.

According to our research, all categories of clustering mechanisms are based on different specific network scenarios or assumptions. That is to say, there is no one single clustering algorithm which can be satisfied in all kinds of ad hoc network circumstances. They may perform better on some specified network environments but worse on others. Obviously, no clustering algorithm is a panacea which is suitable for all ad hoc network scenarios because different algorithms have different researching backgrounds or footholds. Furthermore, a lot of clustering techniques can not be treated as mature commercial products which can be used into realistic applications since most algorithms were developed in laboratories using simulation tools. Fortunately, the studies on clustering mechanisms have been flourishing and this trend will continue along with more and more widely used ad hoc network applications. We believe that the clustering techniques will be commercially mature in the near future under the prosperous development of wireless communication and less hardware constraints of mobile devices.

We propose a new clustering algorithm named K-hops Passive Clustering Algorithm (KHPCA). This algorithm is based on the passive clustering algorithm and mainly focuses on two key considerations of clustered ad hoc network. One is the size of the cluster. If the cluster size were too small, the movement of nodes would cause duplicate routing and most frequent routing information exchange within cluster or among clusters. Thus, this is fatal for the mobile nodes which have weak transmission power and limited battery supply. On the other hand, if the size were too big, it would cause great network latency even transmission interruption. Hence, we use **K** as a variable to define a proper cluster size according to the different network scenarios. The other consideration is the heavy data traffic load of clusterhead that causes high overheads in information collection and frequent cluster changes due to enormous re-clustering . The basic idea of KHPCA is to change clusterhead dynamically and instantly to keep network stability and to enhance network robustness. Differing from most existing clusterhead selection mechanisms which are based on overall exchange of routing information, we propose the concept of "***Candidate-Clusterheads***". When a new clusterhead is needed, we can just elect the new clusterhead from these candidate-clusterheads by node id instead of whole exchange of routing information. As a result, this mechanism can greatly reduce data traffic, maintain stable network topology and improve the network robustness accordingly.

1.4 Thesis Outline

The organization of the thesis is as follows. In the next chapter, we firstly give a global view of ad hoc networking and its significant characteristics and applications. Then, we introduce the representative routing protocols in ad hoc networks. Meanwhile, we discuss the origin and some basic ideas of clustering mechanisms. After that, we discuss the significance of the research in clustering methodology of ad hoc networks and provide applications and developments of clustering. In the third chapter, the significant clustering algorithms have been classified into seven categories. We give a detailed description and analysis among them and illustrate the merits and constraints of these algorithms. Followed by the fourth chapter, we proposed a new clustering algorithm called **KHPCA** (K-hops Passive Clustering Algorithm). We also introduce the simulation environment and give some simulation results in this chapter. Finally, we describe the open issues and then conclude this thesis.

CHAPTER 2

Clustering in Ad Hoc Networks

2.1 Ad Hoc Network

Wireless computing is a rapidly emerging technology providing users with network connectivity without being tethered off of a wired network. Generally, Wireless networks can be classified in two types: infrastructured networks and infrastructureless (ad hoc) networks.

Infrasturctured network, which is called conventional wireless network, is a wireless network built on-top of a ``wired" network and thus creates a reliable infrastructured wireless network. A mobile host communicates with a bridge in the network (called base station) within its communication radius. The mobile unit can move geographically while it is communicating. When it goes out of range of one base station, it connects with new base station and starts communicating through it. This is called "hand-off". In this approach the base stations are fixed. An example of conventional wireless network is the cellular-phone network where a phone connects to the base station with the best signal quality. When the phone moves out of range of a base-station it does a ``hand-off" and switches to a new base-station within reach. The ``hand-off" should be fast enough to be seamless for the user of the network. Other more recent networks of this kind are wireless networks for offices, cafés, etc, which usually are called Wireless Local Area Networks (WLAN).

In contrast to conventional wireless networks, infrastructureless wireless network, which is more commonly called an ad hoc network, does not rely on pre-existing network infrastructure but allows direct communication between the mobile stations. Each mobile station can act as a relay or route to forward traffic toward its destination. Ad hoc networks are self-organizing, mobile wireless communication networks. They operate in a completely distributed manner and are independent of pre-existing network infrastructure. Figures 2.1 and 2.2 are the drawings of conventional wireless networks and ad hoc network respectively. BSS in these figures is defined as a group of stations that are under the direct control of a single coordination function.

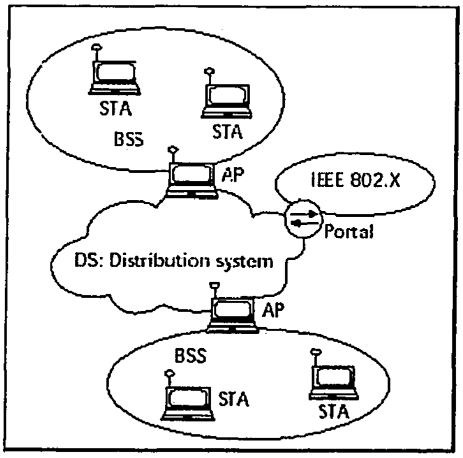


Figure 2.1: Sketch of an infrastructure network

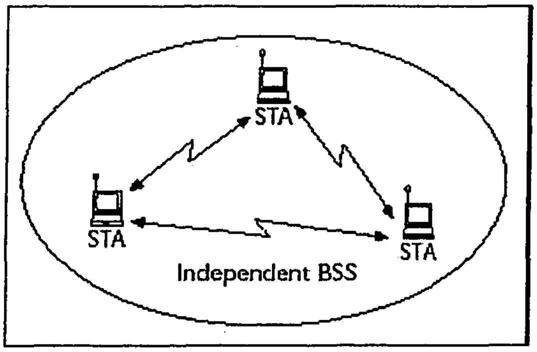


Figure 2.2: Sketch of an ad hoc network

Ad hoc networks have significant advantages over conventional wireless networks. Some of these advantages are:

- Use of ad hoc networks could increase mobility and flexibility because ad hoc networks can be brought up and torn down in very short time.
- Ad hoc networks could be more economical in some cases as they eliminate fixed infrastructure costs and reduce power consumption at mobile nodes.
- Ad hoc networks are more robust than conventional wireless networks because of their non-hierarchical distributed control and management mechanisms.
- Because of short communication links (node-to-node instead of node to a central base station), radio emission levels could be kept at low level. This increases the possibility of spectrum reuse or even the possibility of using unlicensed bands.
- Because of multi-hop support in ad-hoc networks, communication beyond Line Of Sight (LOS) is possible at high frequencies.

2.1.1 Definition and Characteristics of Ad Hoc Networks

Ad hoc is a Latin phrase which means "for this". It generally signifies a solution that has been tailored to a specific purpose. In computer networking, "Ad hoc" indicates a specific wireless network system. This system emphasizes the concept of multi-hop, self-organizing and no central administration. Simply speaking, ad hoc networks are formed dynamically by an autonomous system of mobile nodes that are connected via wireless links without using the existing network infrastructure or centralized administration such as base stations (BSs) or access points (APs).

The fundamental property of ad hoc networks is that node mobility causes the network topology to be changed continuously. The transmission range of nodes is neither fixed nor symmetric. Consequently, the links of an ad-hoc network are not fixed entities. Their status changes over time and is dependent on the relative spatial location of the nodes, transmitter and receiver characteristics, and the signal propagation properties of the environment. Thus, each node acts both as a router and as a host. As nodes move freely and independently, the topology of an ad hoc network changes dynamically and arbitrarily.

The lack of a fixed network and the mobility of the nodes lead to two important features of ad hoc networks, namely, multi-hop packet routing and mobile routers. By contrast with infrastructured networks, ad hoc networks cannot rely on dedicated base stations and routers to forward traffic across fixed network segments between mobile users. Furthermore, mobile nodes can not communicate with each other directly because of limited transmission range and node mobility. Therefore, store-and-forward packet routing is required over multiple-hop wireless paths. That is why the literature sometimes uses the term multi-hop network for an ad hoc network. Consequently, the mobile nodes themselves must cooperate in order to dynamically maintain routes and forward traffic on behalf of other nodes, that is, the mobile nodes themselves must be routers. In order to maintain communications and the subsequent dynamic status of the wireless network links, the routers must implement adaptive algorithms that are responsive to dynamics in the network topology, without over-utilization of network resources.

According to the above viewpoints, compared with other computing network systems, ad hoc networks have the following unique characteristics [17],[18],[24],[29]:

- ♦ *Autonomous and infrastructureless.* Ad hoc networks can be set up instantly at any time and any place without existing fixed infrastructure. There is no centralized administration entity that is required to manage the operation of the different mobile nodes. Each node operates in distributed peer-to-peer mode, acts as an independent router and generates independent data. Network management has to be distributed across different nodes, which brings added difficulty in fault detection and management.
- ♦ *Multi-hop routing.* There is no default router available in ad hoc networks, every node acts as a router and forwards each others' packets to enable information sharing between mobile hosts.
- ♦ *Dynamic topology.* Nodes are mobile and can be connected dynamically in an arbitrary manner. Links of the network vary in time and are based on the proximity of one node to another node, resulting in route changes, frequent network partitions, and possibly packet losses.
- ♦ *Bandwidth constrained.* Wireless links have significantly lower capacity than the wired links. They are affected by several error sources that result in degradation of the received signal and high bit error rate.
- ♦ *Energy constrained operation.* Because batteries carried by each mobile node have limited power supply, processing power is limited, which in turn limits services and applications that can be supported by each node. This becomes a bigger issue in mobile

ad hoc networks because, as each node is acting as both an end system and a router at the same time, additional energy is required to forward packets from other nodes. Therefore, the most important system design criteria for optimization may be energy conservation.

- *Limited physical security.* In ad hoc networks, mobility implies higher security risks than static operation because portable devices may be stolen or their traffic may cross insecurely wireless links. Eavesdropping, spoofing and denial-of-service attacks should be considered.
- *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.
- *Fluctuating link capacity.* The effects of high bit-error rates might be more profound in a multi-hop ad hoc network, since the aggregate of all link errors is what affects a multi-hop 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.
- *Limited network scalability.* At present, network management algorithms are mostly designed to work on fixed or relatively small wireless networks. Many mobile ad hoc network applications involve large networks with tens of thousands of nodes such as sensor networks and tactical networks. Therefore, scalability is critical to the successful deployment of these networks.

♦ *Transitory network lifetime*. Establishment of an ad hoc network must based on a specific scenario and the network system will be removed after deployment. Therefore, compared with fixed network systems, the network lifetime of an ad hoc network is transitory.

2.1.2 Applications and Development of Ad Hoc Networks

A mobile ad hoc network includes several advantages over traditional wireless networks, which include ease of deployment, speed of deployment, and decreased dependence on a fixed infrastructure. Developed for military purpose originally, ad hoc networks have been widely used in military, emergency, and disaster discovery communication. Gradually, non-military applications have also grown substantially. Especially in the past few years, with the rapid advances in mobile ad hoc networking research, mobile ad hoc networks have attracted considerable attention and interests from commercial business industry, as well as the standards community. For example, the introduction of new technologies such as Bluetooth, IEEE 802.11 and Hyperlan greatly facilitates the deployment of ad hoc technology outside of the military domain. In the following, we present some significant applications of mobile ad hoc networks [13].

♦ *Tactical networks*. Needless to say, the technology of ad hoc networking is primarily applied in military applications, such as battlefield in an unknown territory where an infrastructured network is almost impossible to have or maintain. In such situations, the ad hoc networks having self-organizing capability can be effectively used where other technologies either fail or can not be deployed effectively. Therefore, an ad hoc network is the first choice of automated battlefield communication because it is less dependence on a fixed infrastructure, fast of deployment and anti-destructibility.

- *Emergency-management applications.* Responding to emergency situations such as disaster recovery is another naturally fitting application in the ad hoc networking domain. For instance, Restoring communications quickly is essential under the natural disasters where the entire communications infrastructure is in disarray. By using ad hoc networks, an infrastructure could be set up in hours instead of days or weeks required for wired communications.
- *Sensor Networks.* In home applications, smart sensor nodes and actuators can be buried in appliances to allow end users to manage home devices locally and remotely. Besides, environmental applications include tracking the movements of animals (e.g., birds and insects), chemical or biological detection, precision agriculture, etc.
- *Collaborative work.* Mobile conferencing is without a doubt one of the most recognized applications. For some business environments, Establishment of an ad hoc network is essential for mobile users where they need to collaborate in a project outside the typical office environment.
- *Personal Area Networks and Bluetooth.* A Personal Area Network (PAN) creates a network with many devices that are attached or carried by a single person. Even though the communication of devices within PAN does not concern mobility issues, the mobility becomes essential when different PANs need to interact with each other.

Ad hoc networks provide flexible solutions for communications among different PANs. For example, Bluetooth can provide a wireless technology built-in to current PDAs and up to eight PDAs, called *piconet* that can exchange information.

- *Home and enterprise networking.* The wireless computers at home or at a building can create an ad hoc network where each mobile user can communicate with the others without knowing their original places.
- *Embedded computing applications.* Several ubiquitous computing internet-working machines offer flexible and efficient ways of establishing communication methods with the help of ad hoc networking. Many mobile devices already have add-on inexpensive wireless components, such as PDAs with wireless ports and Bluetooth radio devices.
- *Educational applications.* Ad hoc networking technology can be used into setting up virtual classrooms or conference rooms. Furthermore, we can set up ad hoc communication during conferences, meetings or lectures.
- *Commercial environments.* Ad hoc networks can be largely used for commercial purposes, such as electronic commerce, business and vehicular services.

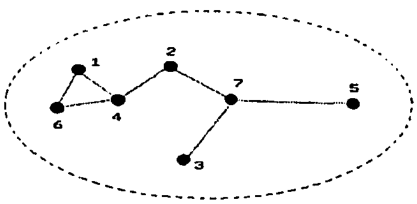
Actually, the applications of ad hoc networks are far more than we illustrated above. With the rapid development of wireless technology and blooming commercial demands, we will find out that ad hoc networking technology can be used more and more widely in a lot of environments.

2.1.3 Architecture of Ad Hoc Networks

Basically, there are two approaches to provide network connectivity in a mobile ad hoc network which are flat-routed architecture (figure 2.3) and hierarchical network architecture (figure 2.4, 2.5) respectively [20].

In flat-routed architecture, all the nodes are identical in terms of responsibility, and there is no concept of special gateways. The mobile nodes are "equal" and the packet routing is done based on peer-to-peer connections, restricted only by the propagation conditions. On the other hand, in hierarchical networks, there are at least two tiers; on the lower tier, nodes in geographical proximity create peer-to-peer networks. In each one of these lower-tier networks, at least one node is designated to serve as a "gateway" to the higher tier. These "gateway" nodes create the higher tier network. Similarly, the "gateway" nodes in that higher tier network create the next higher tier network, and so forth. Although routing between nodes that belong to the same lower-tier network is based on peer-to-peer routing, routing between nodes that belong to different lower-tier networks is through the gateway nodes.

Generally, the network is divided into different clusters in a hierarchical network. Each cluster consists of a clusterhead and many cluster members. These clusterheads create the higher level network. Furthermore, the clusterheads have responsibilities to relay inter-cluster data traffic. Clusterheads can be designated in advance or selected by some specific algorithms. Specifically, a hierarchical network architecture can be divided into two types: single frequency hierarchical architecture and multiple frequency hierarchical architecture. In single frequency hierarchical architecture (figure 2.4), all nodes use one frequency to communicate with each other. Communication between clusterheads is through gateway nodes. However, in multiple frequency hierarchical architecture, each network level needs different communicating frequencies. Nodes in higher level network are also in lower level networks. Therefore, they need different communicating frequencies to establish communication of different network levels. Figure 2.5 is an example of a two-levels hierarchical network architecture. Clusterheads have two communicating frequencies, one is used to communicate between clusterhead and cluster members, the other is used to communicate between clusterheads. Every node in a hierarchical network architecture can be clusterhead. Thus, a proper algorithm for clusterhead election is needed.



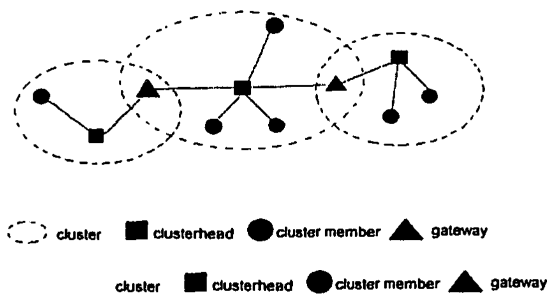


Figure 2.3: Flat-routed network architecture

Figure 2.4: Single frequency hierarchical network architecture

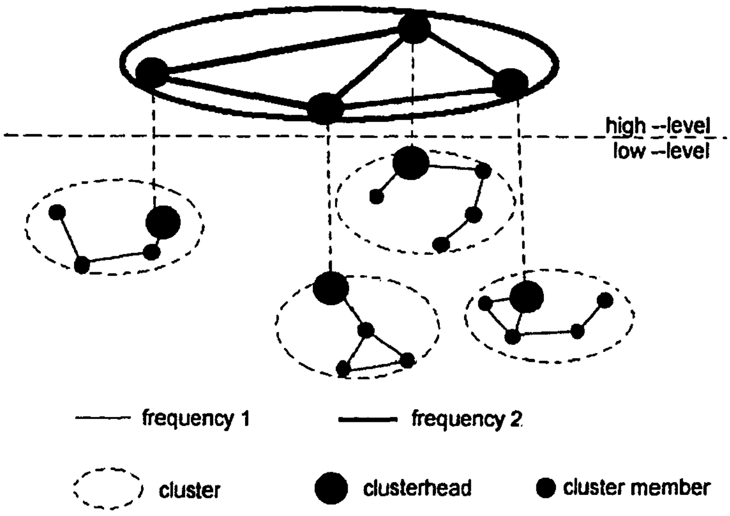


Figure 2.5: Hierarchical network architecture (two-levels)

Flat-routed network architecture is simpler because each node is treated equally. Theoretically, there is no bottleneck in this architecture, Hence, the network is robust. The disadvantage of flat-routed architecture is its poor scalability since each mobile node has to participate in the forwarding and receiving of packets depending on the implemented routing scheme and it introduces significant control overhead to maintain dynamic routing. On the contrary, in a hierarchical network architecture, the functionality of cluster members is relatively simple and there is no need for these member nodes to maintain complicated overall routing information. Therefore, this architecture has good scalability. Furthermore, hierarchical network architectures have better capacity for anti-destructibility. Nevertheless, clusterheads may become bottlenecks in hierarchical architectures.

Consequently, a flat-routed network architecture is often adopted when the network is small; whereas a hierarchical network architecture should be considered when the network is larger. For example, The NTDRS (Near Term Digital System) of the United States army adopts the two levels hierarchical network architecture which is illustrated in figure 2.5.

The clustering algorithms we analyze and propose in this thesis are based on hierarchical network architecture.

2.2 Clustering in Ad Hoc Networks

2.2.1 Routing

Routing is the process of information exchange from one node to the other node in a network. An effective routing mechanism is required to establish a smooth transmission across the network. Since the topology of the network is constantly changing, the issue of routing packets between any pair of nodes becomes a challenging task in mobile ad-hoc networks.

The primary attributes for any routing protocol are [25]:

- *Simplicity*: Simple protocols are preferred for implementation in operational networks.
- *Loop-free*: The paths implied from the routing tables of all hosts taken together should not have loops. Looping of data packets results in considerable overhead.
- *Convergence characteristics*: Time required to converge to new routes after a topology change should not be high.
- *Storage overhead*: Memory overhead incurred due to the storage of the routing information should be low.

- *Computational and transmission overhead*: It is particularly important in mobile wireless networks because the bandwidth of a wireless link is limited, and because mobile devices are typically low-power in order to be portable, and hence do not have the resources for many transmissions and lengthy computations.

Traditional routing algorithms that were designed for infrastructured network can not be operated effectively in ad hoc networks. They tend to exhibit their least desirable behavior under highly dynamic conditions. Routing protocol overhead typically increases dramatically with increased network dynamics. If the protocol overhead is unchecked, it can easily overwhelm network resources. In addition, traditional routing protocols require substantial inter-nodal coordination or global flooding in order to maintain consistent routing information and avoid routing table loops. These techniques increase routing protocol overhead and convergence times. Consequently, although they are well adapted to operate in environments where bandwidth is plentiful and the network links are relatively stable, the efficiency of these techniques conflict with routing requirements in ad-hoc networks. It therefore appears that new routing strategies are required for ad hoc networks that are capable of effectively managing the tradeoff between responsiveness and efficiency.

Routing in mobile ad hoc networks was based on shortest-path algorithms [30]. The Distributed Bellman-Ford algorithm is one such algorithm currently in use in routing algorithms [16]. Generally, the ad hoc network routing protocols can be divided into three main categories, namely proactive routing protocols, reactive routing protocols and hybrid routing protocols.

Proactive Routing, also known as table-driven routing, is defined as a strategy in which routes are continuously maintained for all reachable network destinations. This approach requires periodic dissemination of routing updates to reflect the up-to-date state of the network. Proactive protocols are derived from the traditional distance vector and link state protocols commonly used in wired networks. In this mechanism, every node maintains up-to-date routing tables for each node in the network, so every node knows how to reach other nodes. Although the route discovery method is very simplified, the route recovery and maintenance phase becomes much more involved due to propagating the updates to every node in the system. Consequently, if the nodes are highly mobile, the routing table updates become computationally expensive. Different nodes may still have inconsistent routing tables because of incomplete propagation of dominant set changes. Hence, a packet may temporarily enter a routing loop until the update is correctly processed. The Destination-Sequenced Distance-Vector (DSDV) [42] is the most commonly used proactive routing protocol. DSDV is based on the classical Bellman-Ford routing algorithm. Each node maintains a list of all destinations and number of hops to each destination. Each entry is marked with a sequence number. However, DSDV can not support very high rates of mode mobility and is not sufficiently scalable. The Clusterhead Gateway Switch Routing (CGSR) [12], which uses DSDV as the underlying routing scheme, and the Wireless Routing Protocol (WRP) [36] are other examples.

Reactive routing, also known as On-demand routing, is defined as a strategy in which routes are established and maintained on a demand basis—only if they are needed for communications. This approach requires procedures to acquire new routes and to maintain routes following topology changes. In this mechanism, nodes do not maintain routing tables containing the routes to every node in the network. Contrary to proactive protocols, the route discovery mechanism is not as trivial as looking up a route from a current routing table. In most cases, the route discovery is accomplished by broadcasting a route request massage to immediate neighbors, which in turn broadcast to their neighbors, and so on. This is also known as "flooding". When the destination node receives the route request message, it sends back a route reply message to the source node. Since route discovery is initiated by the source nodes, none of the nodes are responsible for maintaining a routing table containing all the routes.

All reactive algorithms which have been proposed for ad-hoc networks specify three common operations or phases:

1. Route creation phase: Search for destination and construct route.
2. Route maintenance phase: Detect path failure due to node mobility and react to re-establish path.
3. Route erasure phase: Delete either an entire path, or a portion of a path following node movement, when it is no longer needed, or continued use of it could result in packet looping.

The Dynamic Source Routing (DSR) [9][22] and the Ad Hoc On-Demand Distance-Vector (AODV) [43][44] are the leading examples of reactive routing protocols. Other examples include the Temporally Ordered Routing Algorithm (TORA) [40], the Associativity-Based Routing (ABR) [50] and the Signal Stability Routing (SSR) [14].

Hybrid Routing is defined as a strategy which selectively applies either proactive or reactive routing techniques based upon either predefined or adaptive criteria. It combines selected characteristics of both reactive and proactive protocols into a single routing protocol. Specifically, it has been suggested that multiple routing algorithms may need to be implemented to operate in different physical or logical domains, or on different time-scales within the same network. Therefore, Hybrid routing techniques have been advocated to improve the performance of routing in ad-hoc networks. Zone Routing Protocol (ZRP) [20] is an example of such a hybrid scheme. Figure 2.6 shows the simple categorization of ad hoc routing protocols.

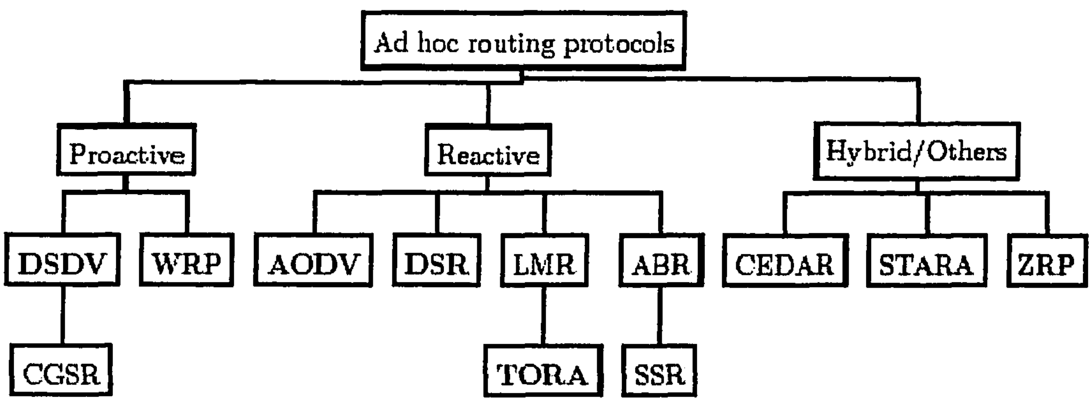


Figure 2.6: Ad hoc network routing protocols

Although numerous routing protocols have been developed for discovering and maintaining routes in ad hoc networks. Each of these protocols utilizes a *flat routing* scheme that is based on flat-routed ad hoc network architecture we described in section

2.1.3 [46]. A flat-routed ad hoc network requires the construction and maintenance of routes that may span a large number of unreliable wireless links. Consequently, in periods of high node mobility, a path may frequently suffer multiple simultaneous failures. Because some action must be taken each time a link break occurs, these protocols have limited scalability in large or rapidly moving networks.

The alternative to these flat routing solutions are hierarchical routing protocols. Hierarchical routing schemes are typically utilized by clustering protocols. Network architects have attempted to devise strategies that dynamically organize ad hoc networks into *clusters*. The cluster topology is then leveraged in order to limit far reaching reactions to topology dynamics. However, unlike the cluster organization of an infrastructured network, the assignment of nodes to clusters in ad hoc networks must be a dynamic process. In a clustered ad hoc network, the network is dynamically organized into partitions called clusters with the objective of maintaining a relatively stable effective topology. The membership and characteristics of each cluster may change dynamically over time in response to node mobility and is determined by the criteria specified by the clustering algorithm. Clustering in ad hoc networks can be used to achieve several different objectives, namely, to support hierarchical routing, to make the route search process more efficient for reactive protocols, to support hybrid-routing in which different routing strategies operated in different domains, or levels of a hierarchy, or to provide more control over access to transmission bandwidth.

2.2.2 Clustering Mechanism

A wireless ad hoc network consists of nodes that move freely and communicate with each other using wireless links. Ad hoc networks do not use specialized routers for path discovery and traffic routing. One way to support efficient communication between nodes is to develop a wireless backbone architecture; this means that certain nodes must be selected to form the backbone. Over time, the backbone must change to reflect the changes in the network topology as nodes move around. The algorithm that selects the members of the backbone should naturally be fast, but also should require as little communication between nodes as possible, since mobile nodes are often powered by batteries. One way to solve this problem is to group the nodes into clusters.

Clustering in ad hoc networks can be informally defined as "grouping of nodes into a manageable set". The concept of dividing the geographical region to be covered into small zones has been presented implicitly as *clustering* in the literature [38]. Clustering algorithms and respective protocols are applied in communication networks to organize all nodes into groups and to obtain a hierarchical network organization. This is done because in large networks a completely flat network structure would pose several problems. For example, routing tables and location registers would grow to an immense size. Also, each router would have to maintain an immense topology knowledge with relevant characteristics. Its memory and processing power would be exceeded, and table lookup times would increase. Clustering in conventional networks with a wired backbone is a rather static process and can be achieved "offline". In contrast to this, in ad hoc networks the association between a node and its cluster must be a dynamic, adaptive, and self-organizing process. A clustering algorithm is running in a distributed and unsynchronized manner in each node. While the primary goal of clustering in conventional infrastructured networks is to make large networks look smaller (scaling reasons, hierarchical routing), an additional and very important goal of clustering in ad hoc networks is to make a very dynamic topology look less dynamic.

Generally, a cluster consists of one clusterhead and many member nodes. Any node can become a clusterhead if it has the necessary functionality, such as processing and transmission power. Nodes register with the nearest clusterhead and become members of that cluster. Clusters may change dynamically due to the dynamic nature of the mobile nodes. Their association and dissociation to and from clusters disturb the stability of the network and thus reconfiguration of clusterheads is unavoidable. Furthermore, the frequent changing of clusterhead results in adverse influences in the performance of other protocols such as scheduling, routing and resource allocation that rely on it. Thus, optimal selection of clusterheads and partitioning of the nodes into clusters are essential aspects of mobile ad hoc networks. In another words, a clustering algorithm is basically a multi-leader election problem.

A cluster is similar to a "group" in distributed systems. The nodes within the cluster are considered the members of the clusterhead of that particular cluster. To be a member of a clusterhead, a node has to lie within the transmission range of that clusterhead. At the same time, it is also possible that even if a node is within the transmission range of a clusterhead, it may not belong to that clusterhead. On the other hand, if a node falls within the transmission range of more than one clusterhead, it is usually called a *gateway* node. More precisely, a *distributed gateway* is a pair of neighboring nodes from different clusters physically located nearest to each other even though their clusters do not overlap. Most clusterhead election algorithms require that each node is a member of one clusterhead and no two clusterheads are adjacent to each other. That is, two clusterheads can not be one-hop neighbors. Hence, multiple clusters and multiple clusterheads are existent in ad hoc networks. The basic clustering network architecture is shown as figure 2.7.

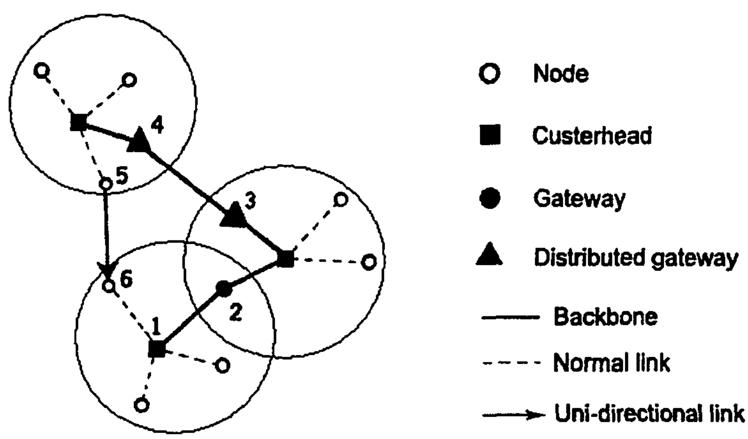


Figure 2.7: Basic clustering network architecture

In ad hoc networks, clustering is crucial for controlling the spatial reuse of the shared channel, for minimizing the amount of data to be exchanged in order to maintain routing and control information in a mobile environment, as well as for building and maintaining cluster-based virtual network architectures. Use of clustering mechanisms can enhance network manageability, scalability and channel efficiency. The first advantage of clustering is stabilizing network topology. By judicious choice of the clusters, we can guarantee a relatively stable cluster topology. With clustering, only the cluster changes and a fraction of gateway changes will impact the network topology changes, and, as a result, routing and connectivity procedures. The second advantage is channel economy. By assigning different channels/codes to different clusters, we can achieve spatial channel/code reuse. The third advantage is ease of location management. Clustering is also known to be useful in improving the MAC resource management. The last advantage of clustering is that it can provide a simple and feasible power control mechanism for an ad hoc network [27]. The disadvantage of a clustered network organization is that additional signaling traffic is needed in order to maintain the cluster structure. Moreover, some nodes have additional workload such as processing, aggregation of topology information and packet forwarding and might become a bottleneck in the network system.

2.3 Measurements and Definitions of Clustering Algorithms

Generally, quality measures of clustering algorithms may be categorized into those that measure the dynamic properties of the protocol (e.g., cluster stability and signaling overhead), and those that measure the performance at the packet level (e.g., delay and throughput) [35]. The basic criteria for measuring cluster algorithms are organized as follows:

- *Direct access*: Every non-clusterhead node should have direct access to at least one clusterhead, so to make possible, for example, "virtual" operation at a higher level.
- *Independence*: No two clusterheads can be neighbors in the network. In graph theoretic terms, the set of clusterheads should form an independent set in the graph, i.e., a set of vertices in which no two vertices are connected by an edge. This is motivated by the need to cover the network with a "well scattered" set of clusterheads, so that each node in the network has a clusterhead in its neighborhood and it has a direct access to that clusterhead.
- *Speed and simplicity*: The algorithm should be very fast and simple, to generate negligible overhead, to adjust quickly to changing conditions and to be a practical solution for configuring the clusters.
- *Distributed operation*: It is motivated by the mobile, decentralized nature of the ad hoc networks. An efficient algorithm should find the clusterheads in a distributed way, without centralized control or need for global information.

Universal definitions of clustering algorithms are :

Definition 1: An ad hoc network can be modeled as an undirected graph G=(V, E) in which V, |V| = n, is the set of nodes and E is the set of edges. There is an edge {u,v} $\in$ E if and only if u and v can mutually receive each others' transmission. In this case we say that u and v are ***neighbors***. (1-hop neighbors)

Definition 2: The ***distance*** of two nodes d(x,y) is the minimum reachable hops between node x and y.

Definition 3: A cluster $C_i \in V$ consists of a set of nodes. For any nodes x and y ($x,y \in C$), if $d(x,y) < A$ (A is a variable system parameter) and if $C_i \cap C_j = \varnothing$ ($i \neq j$), C_i and C_j are called ***nonoverlapping clusters***, else they are ***overlapping clusters.***

Definition 4: A ***dominating set*** is a group of all clusterheads within network. The base of a dominating set is the number of clusterheads. In a dominating set, if any two clusterheads are not neighbors, this dominating set can be called ***irrelative dominating set.***

Definition 5: the number of node c's 1-hop neighbors is defined as c's degree.

Definition 6: For any cluster which has clusterhead, if the maximum distance between clusterhead and any nodes within this cluster is A, this cluster is defined as A-hops cluster with clusterhead. For any cluster which has no clusterhead, if any node within this cluster can reach each other in less than A hops, this cluster is defined as A-hops cluster without clusterhead. The cluster is called clique cluster if A equals to 1.

In this thesis, we quote the common assumptions which were used in most clustering algorithms. Firstly, nodes within an ad hoc network have equal transmission power. Secondly, each node has knowledge of its neighboring node's sequence number as soon as the network is initiated. In addition, a message sent by a node is received correctly within a finite time by all its 1-hop neighbors. Finally, the network topology does not change during the algorithm execution.

2.4 Applications and Development of Clustering

Clustering algorithms are applied in communication networks to organize all entities into groups and to obtain a hierarchical network organization. Such methods are employed in internet–based networks, cellular networks, and ATM networks using the Private Network–to–Network Interface (PNNI). Furthermore, Clustering algorithms are an essential part in the protocol suite of ad hoc networks. The radio technology Bluetooth, for example, implements a clustering principle in order to form wireless "piconets" of devices [7].

Although clustering algorithms for ad hoc networks have been investigated broadly recently, most of them were executed through simulation tools or under laboratorial platforms. Therefore, the commercial applications of clustering algorithms are so rare nowadays. However, the development of wireless communication is very expeditious. Consequently, under the rapidly increasing commercial demand of ad hoc networking technology, as an indispensable part of ad hoc network, clustering mechanisms will be involved into more and more practical fields.

CHAPTER 3

Broad Research and Analysis of Existing Clustering Algorithms

Basically, there are mainly two opposite approaches of clustering algorithms for management of ad hoc networks. One approach eliminates the requirement for a clusterhead altogether when forming the cluster. This mechanism is also called non-clusterhead clustering algorithm. The other approach is based on the concept of clusterhead. That is, selecting a central node (clusterhead) in every cluster.

The former approach is to have all nodes maintain knowledge of the network and manage themselves [31][38]. This circumvents the need to select leaders or develop clusters. However, it imposes a significant communication responsibility on individual nodes. Each node must dynamically maintain routes to the rest of the nodes in the network. With large networks the number of messages needed to maintain routing tables may cause congestion in the network. Ultimately, this traffic will generate huge delays in message propagation from one node to another.

The latter approach is to identify a subset of nodes within the network and vest them with the extra responsibility of being a leader (clusterhead) of certain nodes in their proximity. The clusterheads are responsible for managing communication between nodes in their own neighborhood as well as routing information to other clusterheads in other neighborhoods. Typically, backbones are constructed to connect neighborhoods in the network. Past solutions of this kind have created a hierarchy where every node in the network was no more than one hop away from a clusterhead. In large networks this approach may generate a large number of clusterheads and eventually lead to the same problem as stated in the first design approach. Therefore, it is desirable to have control over the clusterhead density in the network.

Admittedly, clusterheads are totally different from base stations in infrastructured network. They have no support of wired networking architecture and have to be elected dynamically by a particular algorithm. Furthermore, they do extra work with respect to ordinary nodes, and therefore they may become the bottleneck of the cluster. Thus, we can adopt non-clusterhead clustering algorithm when the network is small or the density of nodes is high and the transmission power is relatively low. This permits us to avoid vulnerable centers and hot spots of packet traffic flow. However, as mentioned before, it will cause congestion in the network and the heavy traffic will generate huge delays in message propagation if every node has the responsibility to maintain the routing table. As a result, most of existing clustering schemes are based on the concept of clusterhead.

In this chapter, we discuss one significant non-clusterhead clustering algorithm in Section 3.1. Then, we mainly investigate and analyze the clustering algorithms that are based on clusterhead in the latter sections.

3.1 Adaptive Clustering Algorithm

In order to overcome that a clusterhead may become the bottleneck of the cluster, Adaptive Clustering Algorithm [21][31][32] adopts a fully distributed approach for cluster formation and intra-cluster communications. The objective of this clustering algorithm is to find an interconnected set of clusters covering the entire node population. That is, the system topology is divided into small partitions which are named clusters, with independent control. In the network topology, each node within a cluster is treated equally since there is no notion of clusterhead. This permits us to avoid vulnerable centers and hot spots of packet traffic flow.

This approach is intended to provide controlled access to the bandwidth and scheduling of the nodes in each cluster in order to provide QoS support. Hierarchical routing and path maintenance was a secondary concern. The proposed algorithm is very simple and uses node ID numbers to deterministically build clusters of nodes that are reachable by two-hop paths. Specifically, clustering is performed in two phases: initial cluster setup and cluster maintenance. It is assumed that the nodes do not move during setup. The nodes with the highest weight among their neighbors start the creation of clusters. Afterward, the other nodes will decide to join a certain cluster. A node broadcasts its final decision only after its neighbors with higher weights already made a decision. Whenever the 2-hop condition, which means that any two nodes are at most 2 hops away in a cluster, can not be satisfied, a maintenance scheme will be applied. This scheme was designed to minimize the number of node transitions from one cluster to another. The algorithm produces an incomplete clustering structure until the algorithm finishes (in both setup and maintenance). Once the structure is decided, it is optimized.

The algorithm uses assumptions common to most radio data link protocols. Firstly, every node has a unique ID and knows the IDs of its 1-hop neighbors. Then, a message sent by a node is received correctly within a finite time by all its 1-hop neighbors. Finally, network topology does not change during the algorithm execution. Within each cluster, nodes can communicate with each other in at most two hops. The clusters can be constructed based on node ID. Designated by T_x the set of ID numbers of node x and x's 1-hop neighbors. The node ID of x is X_{id}. This distributed clustering algorithm can be described as follows.

A. If X_{id} is smaller than any node's ID within T_x, XC_{id},which is the cluster ID of x, is designated as X_{id}. Then, node broadcasts cluster information and X_{id} is eliminated from set T_x.

B. When receiving broadcast cluster information, node y sets node's cluster ID as XC_{id}. If X_{id} equals XC_{id} and YC_{id} is larger than XC_{id} or YC_{id} is unknown, YC_{id} is designated as XC_{id}. X_{id} is eliminated from set T_x; if Y_{id} is the smallest ID number of T_x, and YC_{id} is unknown, YC_{id} is designated as Y_{id}. Y broadcasts cluster information.

C. Algorithm is terminated when T_x is empty.

In this algorithm, each node only broadcasts one cluster message before the algorithm stops, and the time complexity is O(|V|) where *V* is the set of nodes. The clustering algorithm converges very rapidly. In the worst case, the convergence is linear in the total number of nodes.

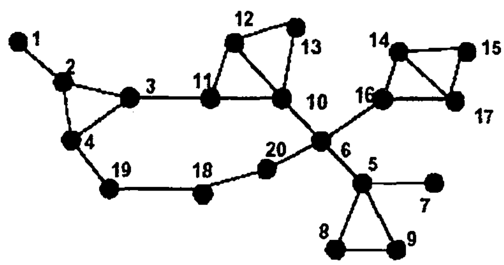


Figure 3.1(a): Network topology

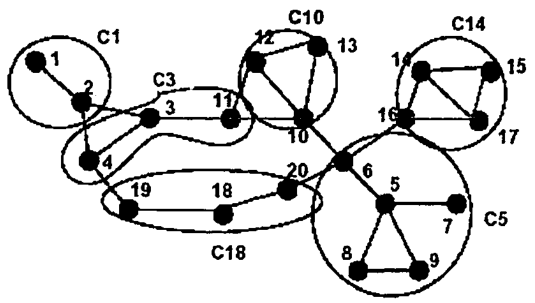


Figure 3.1(b): Clustering

Figure 3.1(a) and (b) show an example of a network topology before and after the algorithm execution.

Simulation shows that this algorithm is robust because its simple cluster structure without clusterhead. However, the major drawback of this algorithm is that it assumes a stable topology during cluster setup. In a mobile scenario, this may lead to a loss of messages, since the topology assumed by the algorithm is not valid anymore. On the other hand, the number of signaling messages sent during cluster establishment is limited to the number of nodes N, and the convergence time is bounded by O(N). Therefore, this algorithm is only suited for quasi-static ad hoc networks.

3.2 The Lowest-ID Clustering Algorithm

The Lowest-ID clustering algorithm was proposed in [15] and was revisited later in the "multimedia multi-hop mobile networks" context [51]. The Lowest-ID, also known as identifier-based clustering, is one of the most popular clustering schemes used in the old as well as recent ad hoc networks literature. This scheme 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.

The Lowest-ID algorithm proceeds as follows and results in the formation of clusters which are at most two hops in diameter.

- Each node is given a distinct ID and it periodically broadcasts the list of its neighbors (including itself).

- A node which only hears nodes with ID higher than itself is a "clusterhead".
- 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.

The following figure 3.2 shows a ten-node example based on Lowest-ID clustering algorithm. Node 1,2 and 4 are clusterheads and nodes 8,9 are gateway nodes.

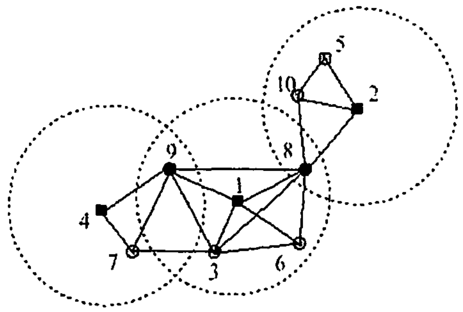


Figure 3.2: Example of cluster formation of Lowest-ID Clustering algorithm

3.3 The Highest-Degree Clustering Algorithm

The node degree is another commonly used heuristic in which nodes with higher degrees are more likely to become clusterheads. The Highest-Degree, also known as connectivity-based clustering, was originally proposed by Gerla and Parekh [39],[51], 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. In the case of tie, the node's Lowest or Highest id may be used. 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).

This approach can result in a high turnover of clusterheads if the network topology changes [51]. This is undesirable because of the high overhead associated with clusterhead changes. Data structures have to be maintained for each node in the cluster. As new clusterheads are elected, these data structures must be passed from the old clusterhead to the newly elected clusterhead. Re-electing clusterheads could minimize this network traffic by circumventing the need to send these data structures.

The Highest-Degree algorithm proceeds as follows:

- Each node broadcasts the list of nodes 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 5 shows the same ten-node example, where nodes 5,7 and 8 are clusterheads;

nodes 2,3,9,10 are gateway nodes.

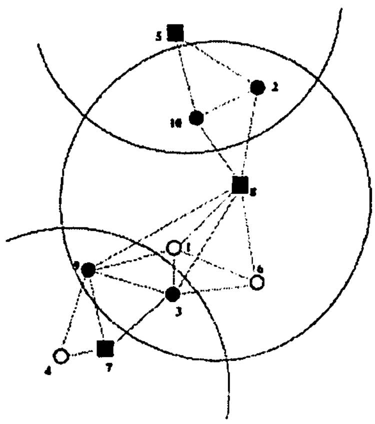


Figure 3.3: Example of cluster formation of Highest-Degree Clustering algorithm

3.4 The Node-Weight clustering algorithm

[5],[6],[11] and [52] are typical examples of this kind of algorithm. In this approach, each node is assigned weights (a real number ≥0) based on its suitability of being a clusterhead. A node is chosen to be a clusterhead if its weight is higher than any of its neighbor's weight; otherwise, it joins a neighboring clusterhead. The smaller node id is chosen in case of a tie.

Generally, Node-Weight clustering algorithms should satisfy the following three ad hoc networks clustering properties. Firstly, every ordinary node has at least a clusterhead as neighbor (dominance property). This is necessary to ensure that each ordinary node has direct access to at least one clusterhead, thus allowing fast intra- and inter-cluster communications. Secondly, every ordinary node affiliates with the neighboring clusterhead that has the bigger weight. That ensures that each ordinary node always stays with the neighboring clusterhead with the bigger weight, i.e., with the clusterhead that can give it a "guaranteed good" service. Thirdly, no two clusterheads can be neighbors. It guarantees that the network is covered by a "well scattered" set of clusterheads, so that each node in the network has a clusterhead in its neighborhood and it has direct access to that clusterhead.

Lowest-ID clustering algorithm was generalized to weight based clustering schemes, referred to as DCA (Distributed Clustering Algorithm) and DMAC (Distributed Mobility-Adaptive Clustering Algorithm) in [5]. Each node is assumed to have a unique weight instead of just the node ID and the weights are totally ordered. However the technique of assignment of weights has not been discussed. DCA is intended for quasi-static networks in which nodes are slow moving, if moving at all. DMAC is designed for higher mobility network. A major difference between DCA and DMAC is that nodes can move during cluster setup in DMAC. DMAC is adaptive to changes in the network topology, in a way that each node reacts locally to failures of links and appearance of new links, and it changes its status accordingly. DMAC is more adaptive, since it immediately creates cluster after a topology change happens. It builds up clusters faster, but these clusters are not so optimal in the beginning, this structure will be optimized in several iterations. And, thus, more signaling messages are sent until the cluster structure is optimized. Simply speaking, DCA and DMAC has the following properties:

- As for the DCA, the selection of the clusterheads is based on the new weight-based criteria, thus having the possibility to express *preferences* on which nodes are better suited to be clusterheads.

- Nodes can move, even during the clustering set up. DMAC is adaptive to the changes in the topology of the network, due to the mobility of the nodes or to node addition or removal.
- A node decides its own role (clusterhead or ordinary node) solely knowing its current one-hop neighbors.

Furthermore, Weighted Clustering Algorithm (WCA) and Maximal Weighted Independent Set (MWIS) [6] [11] are both representative node-weight clustering algorithms. Basically, each node is assigned a positive real number weight based on certain criteria, such as transmission range. The procedure of clusterhead election can be simplified to find an independent set which includes maximal weighted nodes. [52] proposed a new technique to optimize WCA in that each clusterhead handles the maximum possible number of mobile nodes in its cluster in order to facilitate the optimal operation of the medium access control (MAC) protocol. Consequently, it results in the minimum number of clusters and hence clusterheads. Similarly, Automatic On-demand Weighted algorithm (AOW) tried to optimize WCA in finding more flexible criteria of node weight.

The main advantage of choosing cluster head based on node-weight is that, by representing with the weights mobility-related parameters of the nodes, we can choose for the role of cluster head those nodes that are better suited for that role. Figure 3.4 shows the example of Node-Weight clustering algorithm, where figure 3.4(a) is an ad hoc network with nodes and their weights and figure 3.4(b) is the correct clustering.

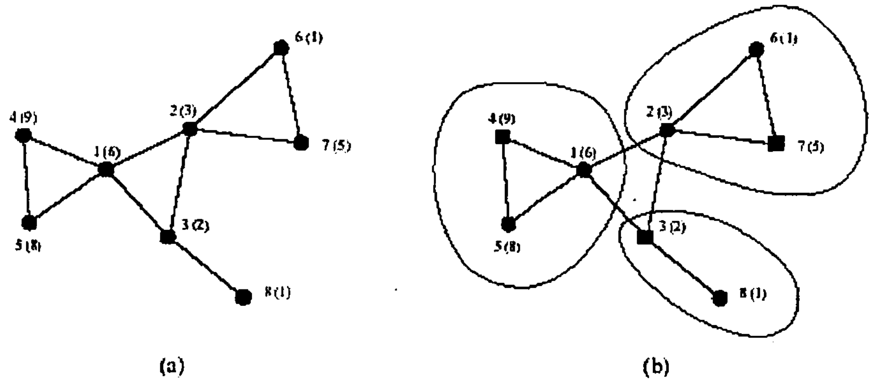


Figure 3.4 Example of node-weight clustering algorithm

3.5 Least Cluster Change (LCC) Clustering Algorithm

Commonly, the most important criterion of a clustering algorithm is keeping network stability. The Lowest-ID clustering algorithm was improved by its variant LCC (Least Clusterhead Change) [12] by imposing an additional rule on the clustering process to achieve stable network topology. That is, if a member (id= *i*.) of cluster C moves within range of another cluster C' with higher id, then do not perform reclustering unless *i* is the clusterhead of C. This helps in drastically reducing the number of changes in clusterheads due to reclustering. This approach results in formation of clusters which are 2 hops in diameter and converges in O(*d*) time, where *d* is the diameter of the network. The clusterhead takes the responsibility of coordinating transmissions of packets and route discovery, and thus the network does not have to depend on classic flooding for routing.

Specifically, the LCC clustering algorithm only considers two conditions which cause the cluster head to change. One is when two clusterheads come within range of each other, and the other is when a node becomes disconnected from any other cluster. This is an improvement (in stability) over existing algorithms which select the cluster head every time the cluster membership changes. The following is the specification of the LCC algorithm.

1. Firstly, Lowest-ID cluster algorithm or Highest-Degree clustering algorithm is used to create initial clusters.
2. When an ordinary node in cluster i move into a cluster j, no clusterhead in cluster i and j will be changed (only cluster members are changed).
3. When an ordinary node moves out of its cluster and does not enter into any existing cluster, it becomes a new clusterhead, forming a new cluster.
4. When clusterhead C(i) from cluster i moves into the cluster j, it challenges the corresponding clusterhead C(j). Either C(i) or C(j) will give up its clusterhead position according to Lowest-ID or Highest-Degree algorithm or some other well defined priority scheme.
5. Nodes which become separated from a cluster will recompute the clustering according to Lowest-ID or Highest-Degree algorithm.

In case of the high mobility of network topology, the LCC algorithm can reduce the changes of clusterheads. However, the main drawback of LCC is that the cost of clusterhead is relatively large. Furthermore, this algorithm does not consider system load balance and the node's power depletion.

3.6 Max-Min D-Cluster Formation Clustering Algorithm

The general purpose of clustering algorithms is to select certain nodes (clusterheads) to form a wireless backbone. However, most algorithms restricted themselves to 1-hop clusters. That is, no node can be more than one hop away from its clusterhead. Even though this restriction simplified the algorithm execution, it would cause a large number of clusters and heavy traffic load. Max-Min D-Cluster Formation algorithm [2] started with the aim of generalizing the clustering schemes so that a node is either a clusterhead or at most d hops away from a clusterhead, where $d \geq 1$, i.e., a d-hop dominating set. The value of d is a parameter of this algorithm. Furthermore, the design goals of Max-Min D-Cluster Formation are as follows:

1. Nodes asynchronously run the algorithm: no need for synchronized clocks,
2. Limit the number of messages sent between nodes to $\mathbf{O}(d)$,
3. Minimize the number and size of the data structures required to implement the heuristic,
4. Minimize the number of clusterheads as a function of d,
5. Formation of backbone using gateways,
6. Re-elect clusterheads when possible (to keep the network stability),
7. Distribute responsibility of managing clusters is equally distributed among all nodes (to keep the fairness of nodes' responsibilities).

Based on the node id scheme, the mechanics of the algorithm are simple. At some common epoch each node initiates $2d$ rounds of flooding. Each node maintains a logged entry of the results of each flooding round. The rounds are segmented into the first d rounds and the second d rounds. The first d rounds are to propagate the largest node ids. After completion of the first d rounds of flooding the second d rounds of flooding begin, using the values that exist at each node after the first d rounds. The second d rounds of flooding are to allow the smaller node ids to reclaim some of their territory. After completion of the second d rounds each node looks at its logged entries for the $2d$ rounds of flooding. The following three rules explain the logical steps of the algorithm that each node runs on the logged entries. Firstly, each node checks to see if it has received its own original node id in the second d rounds of flooding. If it has then it can declare itself a clusterhead and skip the rest of this phase of the algorithm. Otherwise proceed to the second rule. The second rule is that each node looks for node pairs. Once a node has identified all node pairs, it selects the minimum node pair to be the clusterhead. If a node pair does not exist for a node then proceeds to the third rule. The third rule is to elect the maximum node id in the first d rounds of flooding as the clusterhead for this node.

The most significant merit of this algorithm is the generalization from 1-hop to d-hops. It allows control and flexibility in the determination of the clusterhead density. Furthermore, the number of messages is a multiple of d rounds, providing a very good run time at the network level. Simple data structures have been used to minimize the local resources at each node. Re-election of clusterheads is promoted to minimize transfer of databases and to provide stability. The solution is scalable as it generates a small number of clusterheads compared to some other algorithms. Also, a low variance in cluster sizes leads to better load balancing among the clusterheads. Finally, this algorithm utilizes clusterheads and multiple gateway nodes to form a redundant backbone architecture to provide communication between clusters.

3.7 Passive Clustering Algorithm

In order to solve the cluster limitations of conventional clustering algorithms in a realistic wireless ad hoc network scenario, [19],[26] and [28] proposed solutions to overcome such limitations by using a novel clustering algorithm which is called passive clustering algorithm with a new clusterhead election regulation.

3.7.1 Limitations of Conventional Clustering Algorithms

Unlike in a simulation environment, global information regarding node locations and adjacency relations is hard to collect in a realistic wireless ad hoc network. The major reason of the difficulties comes from unreliable and limited link capacity and from high node mobility. Node locations and neighborhood information are keys for clustering. However, they do vary in time. Therefore, conventional clustering algorithms assume repeated broadcasting of the neighbor list to ensure the correct collection of neighborhood information.

The quasi-stationary assumption is necessary for stable operation in conventional clustering algorithms in which clusterhead election rules are ID driven and weight driven. In conventional algorithms (e.g. Lowest-ID, Highest-Degree), this assumption must hold during the neighborhood information collecting period, cluster initialization, and the re-clustering or clustering maintenance period. The reason for the quasi-stationary assumption is obvious and straightforward. If a node moves, the neighborhood information we collect is stale. Furthermore, node mobility results in adjacency relation changes, which may trigger re-clustering throughout the network.

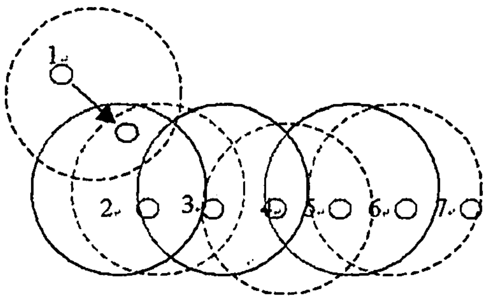


Figure 3.5: Chain reaction

Figure 3.5 shows the case we described above. The dotted circles represents the results of Lowest-ID clustering algorithm while node 1 is out of node 2's transmission range. When node 1 moves to node 2's transmission range, node 2 abdicates its role of clusterhead, and triggers a chain reaction causing a change in all clusterheads. Besides the quasi-stationary assumption, conventional clustering algorithms have to use more assumptions such as uniform transmission power and perfectly coordinated communications to ensure complete collection of neighborhood information.

Another cluster limitation of conventional clustering algorithms is the isolation problem, which means cluster disconnection while there is a feasible radio path. Figure 3.6 shows an example of this problem. Through Lowest-ID clustering algorithm, nodes 1,2 and nodes 3,4 form two clusters respectively which are indicated by two solid circles. However, these clusters can not communicate with each other. If node 2 were elected clusterhead, and node 3 became a gateway, the four nodes would be connected in the clustered structure. With conventional clustering algorithms such as Lowest-ID, the isolation problem can be solved only by using ad hoc extensions of the basic algorithm. For instance, node 2 and 3 become distributed gateways and jointly provide the required connectivity.

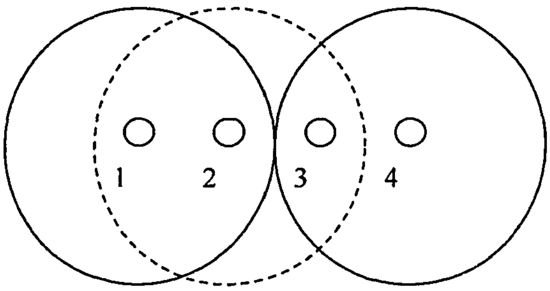


Figure 3.6: Isolation problem

3.7.2 Description of Passive Clustering Mechanism

Conventional clustering algorithms require all participating network nodes to advertise neighbor information repeatedly. Consequently, the overhead is very expensive where bandwidth is limited. Furthermore, conventional clustering algorithms require an initial clustering phase prior to any network layer activities. With passive clustering, we can avoid both of the above disadvantages.

Basically, Passive Clustering is a cluster formation protocol, which does not use dedicated, protocol specific control packets or signals. The pivotal idea of clustering scheme in passive clustering is to opportunistically exploit the neighbor information carried by data packets, without the need of explicit periodical control signals. It does not require an initial clustering phase before data transmission. Therefore, it avoids large number of overheads and the cluster infrastructure can be constructed as a by-product of user data packet exchanges. Furthermore, passive clustering can be used with any on-demand routing protocols since it is consistent with the on-demand routing philosophy.

Passive clustering uses data traffic information to accomplish network clustering, which are relatively immune from logical isolation and lack of connectivity. Clustering stability and fast convergence time are the major merits of passive clustering because the new "*first declaration wins*" rule is used in clusterhead election.

With the *first declaration wins* (FDW) rule, the node that sends a packet first, becomes the clusterhead and "rules" the rest of nodes in its clustered area (radio coverage). The new clusterhead election rule does not require re-clustering to maintain the weight criteria. Besides, FDW can handle the following distributed isolation problem: If there is no gateway in a cluster for a timeout period, the clusterhead resigns, and the rest of nodes in the cluster compete for the clusterhead position.

When a node is ready to be a clusterhead, it declares its readiness by sending its clustering state claim in the MAC packets. Since passive clustering does not support explicit control packets or signals of its own, a clusterhead-ready node must postpone its claim until it has data traffic. After a successful transmission from a newly volunteered clusterhead, every node in the radio coverage can learn the presence of a clusterhead by monitoring packets from it and then changes its node state accordingly.

The passive clustering and maintaining procedure is simple and fully distributed. A node starts with "initial" state when it joins a network. Passive clustering resides in MAC layer to utilize the source ID of MAC packets to collect neighbor information, and to advertise the node's clustering state. Generally, there are four possible node's states. These states are included in MAC packets which will cause extra cost. The passive clustering can be summarized as follows:

1. There are four possible states and one internal state: INITIAL, CLUSTERHEAD, ORDINARY_NODE, GATEWAY, and CLUSTERHEAD-READY.
2. Below the MAC layer, a cluster sub layer (CSL) adds two bits of cluster state information on every outgoing MAC packet. If current state of the node is CLUSTERHEAD-READY, it changes state to CLUSTERHEAD before it tags state information. For incoming packets, the CSL strips cluster state information of the sender, and extracts the sender id information, performs passive clustering, then passes the packet to MAC layer.
3. At cold start, every node is in INITIAL state. There is no state change until a node receives a MAC packet. If the sender state is not CLUSTERHEAD, then its cluster state turns into CLUSTERHEAD-READY. The CLUSTERHEAD-READY node will

be a clusterhead if it successfully transmits an outgoing packet before it receives any packets from another clusterhead. If the packet was from another clusterhead, i.e. sender state is CLUSTERHEAD, adds the clusterhead information (id, and reception time) to the node's clusterhead list, and goes into ORDINARY_NODE state.

4. All the nodes in any state other than CLUSTERHEAD maintain neighbor clusterhead list. Whenever a node receives packets from a clusterhead, it updates the clusterhead list. Meanwhile, it checks the number of alive clusterheads. The state of a non-clusterhead node is determined by the number of clusterheads in the list. When the number of clusterheads of a non-clusterhead node goes to 0, the node transits to INITIAL state. If the state of node is CLUSTERHEAD when the node receives a packet from another clusterhead, it goes into ORDINARY_NODE state.
5. Every node collects the neighbor information as the clustering procedure goes. It stores neighbors' id, state, and the idle time. If the idle time goes beyond the timeout threshold, the entry is removed.
6. Without employing an explicit timer, a node examines the freshness of lists (clusterhead, and neighbor lists) whenever CSL is up – sending/receiving MAC packets.

According to its clustering mechanism and unique FDW rule in clusterhead election, there are many advantages of passive clustering. First of all, clustering can be achieved without using protocol specific, explicit control packets or signals. Besides, passive clustering does not need the initial clustering phase to precede the data and communication phase. Furthermore, passive clustering does not require re-clustering to satisfy clustering regulation like Lowest-ID, when the connectivity changes, because of mobility. Finally, in passive clustering, clustering can be done without collecting complete neighborhood information.

3.8 Clustering Algorithms Analysis

In the above sections, we broadly classified the existing representative clustering algorithms into seven categories. Actually, there is interaction between some, most kinds of clustering algorithms are derived and revised from conventional clustering schemes, such as Lowest-ID and Node-Weight. For example, although as a non-clusterhead clustering algorithm, Adaptive clustering algorithm also can be treated as a Lowest-ID mechanism since its clustering scheme is based on node ids. In addition, a node's id is a specific type of node's "weight". Therefore, generally speaking, Lowest-ID and Node-Weight algorithms are both "weight driven". Least Cluster Change clustering algorithm is a revised edition of Lowest-ID. Furthermore, Max-Min D-Cluster Formation Clustering Algorithm is also based on node ids.

Generally, Lowest-ID, Highest-Degree and Node-Weight clustering algorithms were early developed and created the foundation of clustering algorithms. These three schemes were widely investigated and analyzed to achieve further improvements.

Experiments demonstrate that the system has a low rate of clusterhead change but the throughput is low under the Highest-Degree clustering algorithm. Typically, each cluster is assigned some resources which is shared among the members of that cluster on a round-robin basis [33],[34],[51]. As the network topology changes this approach can result in a high turnover of clusterheads. This is undesirable due to the high overhead associated with clusterhead change over. Data structures have to be maintained for each node in the cluster. As new clusterheads are elected these data structures must be passed from the old clusterhead to the newly elected clusterhead. Re-election of clusterheads could minimize this network traffic by circumventing the need to send these data structures.

The system performance of the Lowest-ID algorithm is better compared with the Highest-Degree algorithm in terms of throughput. Since the environment under consideration is mobile, it is unlikely that node degrees remain stable resulting in frequent clusterhead updates. That means, it provides a more stable cluster formation than the Highest-Degree algorithm. However, the drawback of this algorithm is its bias towards nodes with smaller ids which may lead to the battery drainage of certain nodes.

The main advantage of choosing cluster head based on Node-Weight algorithm is that, by representing with the weights mobility-related parameters of the nodes, we can choose for the role of cluster head those nodes that are better suited for that role. Results proved that the number of updates required is smaller than the Highest-Degree and Lowest-ID algorithms. Since node weights were varied in each simulation cycle, computing the clusterheads becomes very expensive and there are no optimizations on the system parameters such as throughput and power control. Under the Node-Weight algorithm, the node has to wait for all the responses from its neighbors to make its own decision to be a clusterhead or an ordinary node. This algorithm does not account for the amount of time that a node may need to wait to receive responses from its neighbors.

Briefly, existing researches tend to be based on Lowest-ID and Node-Weight clustering algorithms because of their merits. As a revised version of Lowest-ID, the LCC algorithm can make the network topology more stable since the changes of clusterheads are less. However, clusterheads will be easier to become a network bottleneck because the cost of clusterhead is relatively high.

Passive Clustering algorithm is a more promising technique. It is established in solving clustering limitations of conventional clustering schemes (Lowest-ID, Node-Weight). Admittedly, passive clustering will cause some network latency and traffic delay because it relies on on-demand routing protocols. However, the advantages overweigh the disadvantages. As a result, the mechanism proposed in this thesis is based on passive clustering scheme.

Overall, there is no clustering algorithm that is omnipotent. Each scheme was based on specified network scenario, assumptions and conditions. Furthermore, these algorithms have their own design goals or standpoints. Therefore, they may have better performance under their specific scenario but have worse performance under other network environments. Along with the progress of wireless techniques, the development of clustering algorithms still has a long way to go.

CHAPTER 4

Proposed Clustering Algorithm--- K-hops Passive Clustering Algorithm (KHPCA)

In this chapter, we propose a new clustering algorithm, K-Hops Passive Clustering Algorithm, which is based on passive clustering philosophy.

4.1 Preliminaries

"Cluster size" and "Data traffic load of clusterhead" are two most important considerations in a clustered ad hoc network.

When we consider using clustering mechanisms into an ad hoc network environment, the spontaneous question which comes into our mind is how to decide the proper cluster size in a network scenario. If the cluster size were too small, the random movement of mobile nodes would cause duplicate routing and enormous or frequent routing information exchange within a cluster or among clusters. Consequently, this is fatal for the mobile nodes that have weak transmission power and limited battery supply. On the other hand, if the size were too big, which means a large number of ordinary nodes (member nodes) are existing in one cluster, the network routing would be very torpid about node movement. Therefore, it would cause great network latency even unpredictable transmission interruption.

Data traffic load of clusterhead is another key factor of a clustered ad hoc network. Cluster formation simplifies topology maintenance in ad hoc networks. However, it has a negative effect on the clusterheads, because a clusterhead drains its energy more quickly than a normal node. Therefore, a clusterhead election algorithm must also consider network instability due to heavy data traffic load of clusterhead. Theoretically, there exists only one clusterhead in a cluster. The clusterhead has the full responsibility of maintaining routing tables of every node within its cluster, besides, it has a heavy burden of relaying a large amount of inter-cluster data traffic. Compared with ordinary nodes, the battery power of the clusterhead will be depleted rapidly. The processing capacity of a clusterhead will climb to the peak instantly. As a result, the clusterhead would become network bottleneck and the clustered network would suffer from unstable network topology as well as high traffic overheads. The impact to the network is fatal. Meanwhile, through the comprehensive analysis of current clustering algorithms in chapter 3, we easily found out that most algorithms were based on 1-hop or 2-hops methodology. Although 1-hop or 2-hops cluster structure is very simple and the implementation is relatively easy, the disadvantages of these simple structures are obvious too. First of all, as we know, clusterheads play a crucial role in a clustered ad hoc network. If clusterhead became disabled unfortunately, the procedure of re-clustering would be time consuming. Clearly, it would produce severe impact to whole network performance. In addition, the small clusters which are produced by 1-hop or 2-hops methodology would cause frequent re-clustering while mobile nodes are freely moved. Hence, many improved schemes were proposed to maintain network stability. However, re-clustering under such schemes still produced severe time-delay and high network overheads. Furthermore, 1-hop or 2-hops would produce excessively independent small clusters. Under this condition, inter-cluster communications and routing information are too complicated and costly, especially in a large ad hoc network. Finally, most existing clustering mechanisms need nodes which when participating in network clustering repeatedly broadcast their neighbor information. As a result, the overhead of collecting neighbor information is huge.

Therefore, based on the above considerations, we plan to alleviate these constraints according to the following two aspects:

1. Our basic design idea is to optimize network performance through controlling clusters size, i.e. changing amount of nodes in one cluster, and electing clusterheads dynamically and efficiently. Most of existing clusterhead election schemes need "overall" exchange of routing information of clusters. That is, firstly, participating nodes collect global routing information, then derive the proper QoS criteria (sequence No, error rates, etc.) according to this information. Finally, they elect the new clusterhead based on these criteria. The procedure is time consuming and costly. In our new algorithm, we propose the concept of "**Candidate-Clusterheads**". When a current clusterhead is disabled and then a new clusterhead is needed, we can just setup a new clusterhead from these candidate-clusterheads by node id instantly instead of tedious clusterhead re-election. As a result, this mechanism can greatly reduce data package exchange of network, and prevent frequent cluster changes due to enormous re-clustering, and then, greatly improve the network robustness and stability.
2. Our new proposed scheme is based on the passive clustering philosophy. Namely, we

utilize the "listen" information to cluster the network. Network clustering can be done without need of the periodical explicit control packets or signals, the initial clustering phase to precede the data and communication phase or the complete neighbor information. The implementation of the algorithm is through modification of AODV routing protocol, which is a leading on-demand routing mechanism and is broadly used in ad hoc network environments.

We will introduce AODV routing protocol comprehensively in the next section. Then, we provide the detailed description of the proposed clustering algorithm---K-hops Passive Clustering Algorithm (KHPCA).

4.2 AODV Routing Protocol

Ad Hoc On-Demand Distance Vector Routing (AODV) is a popular on-demand based routing protocol [10],[43],[44],[47],[48]. It enables dynamic, self-starting, multi-hop routing between participating mobile nodes wishing to establish and maintain an ad hoc network. AODV allows mobile nodes to obtain routes quickly for new destinations, and does not require nodes to maintain routes to destinations that are not in active communication. It is essentially a combination of both DSR (Dynamic Source Routing) and DSDV (Destination-Sequenced Distance-Vector). It borrows the basic on-demand mechanism of Route Discovery and Route Maintenance from DSR, plus the use of hop-by-hop routing, sequence numbers, and periodic beacons from DSDV. This routing algorithm builds on the DSDV algorithm and the improvement is on minimizing the number of required broadcasts by creating routes on an on-demand basis, as opposed to maintaining a complete list of routes as in DSDV algorithm. Generally, AODV makes use of advantages from both Distance-Vector and On-demand.

One distinguishing feature of AODV is its use of a destination sequence number for each route entry. The destination sequence number is created by the destination to be included along with any route information it sends to requesting nodes. Using destination sequence numbers ensures loop freedom and is simple to program. Given the choice between two routes to a destination, a requesting node is required to select the one with the greatest sequence number.

AODV needs to keep track of the following information for each route table entry:

- Destination IP address: IP address for the destination node.
- Destination Sequence Number: Sequence number for this destination.
- Hop Count: Number of hops to the destination.
- Next Hop: The neighbor, which has been designated to forward packets to the destination for this route entry.
- Lifetime: The time for which the route is considered valid.
- Active neighbor list: Neighbor nodes that are actively using this route entry.
- Request buffer: Makes sure that a request is only processed once.

Basically, AODV defined three message types, namely, Route Requests (RREQs), Route Replies (RREPs), and Route Errors (RERRs). When a source node desires to send a message to some destination node and does not already have a valid route to that destination, it initiates a "path discovery" process to locate the other node. It broadcasts a route request (RREQ) packet to its neighbors, which then forward the request to their neighbors, and so on, until either the destination or an intermediate node with a "fresh enough" route to the destination is located. Figure 4.1(a) illustrates the propagation of the broadcast RREQs across the network. AODV utilizes destination sequence numbers to ensure all routes are loop-free and contain the most recent route information. Each node maintains its own sequence number, as well as a broadcast ID. The broadcast ID is incremented for every RREQ the node initiates, and together with the node's IP address, uniquely identifies an RREQ. Along with its own sequence number and the broadcast ID, the source node includes in the RREQ the most recent sequence number it has for the destination. Intermediate nodes can reply to the RREQ only if they have a route to the destination whose corresponding destination sequence number is greater than or equal to that contained in the RREQ.

During the process of forwarding the RREQ, intermediate nodes record in their route tables the address of the neighbor from which the first copy of the broadcast packet is received, thereby establishing a reverse path. If additional copies of the same RREQ are later received, these packets are discarded. Once the RREQ reaches the destination or an intermediate node with a fresh enough route, the destination/intermediate node responds by unicasting a RREP packet back to the neighbor from which it first received the RREQ. Figure 4.1(b) shows the path of RREP to the source. As the RREP is routed back along the reverse path, nodes along this path set up forward route entries in their route tables which point to the node from which the RREP came. These forward route entries indicate the active forward route. Associated with each route entry is a route timer which will cause the deletion of the entry if it is not used within the specified lifetime, because the RREP is forwarded along the path established by the RREQ.

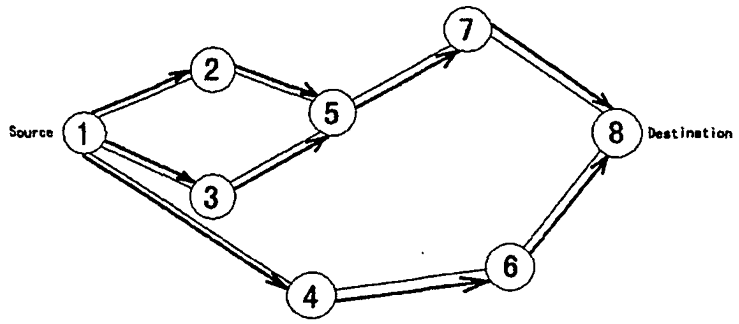


Figure 4.1(a): Propagation of RREQ in AODV

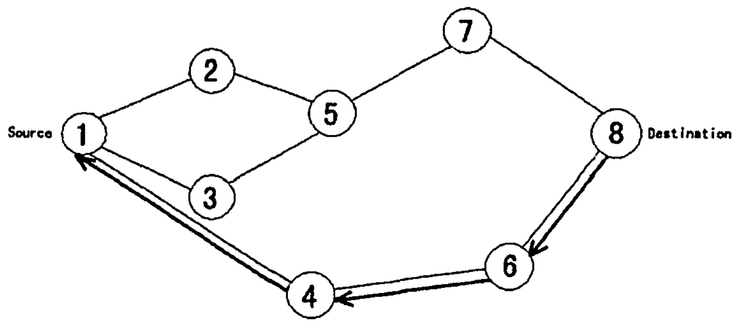


Figure 4.1(b): Path of RREP in AODV

If data is flowing and a link break is detected, a RERR packet is sent to the source of the data in a hop-by-hop fashion. As the RERR propagates towards the source, each intermediate node invalidates routes to any unreachable destinations. When the source of the data receives the RERR, it invalidates the route and reinitiates route discovery if necessary.

An additional aspect of the AODV is the use of "Hello" messages, periodic local broadcasts by a node to inform each mobile node of other nodes in its neighborhood. Hello messages can be used to maintain the local connectivity of a node. However, the use of hello messages is not required. Nodes listen for retransmission of data packets to ensure that the next hop is still within reach. If such a retransmission is not heard, the node may use any one of a number of techniques, including the reception of hello messages, to determine whether the next hop is within communication range. The hello messages may list the other nodes from which a mobile has heard, thereby yielding greater knowledge of network connectivity.

Furthermore, AODV has another two properties. One is that AODV only supports one route for each destination. It should however be fairly easy to modify AODV, so that it supports several routes per destination. Instead of requesting a new route when an old route becomes invalid, the next stored route to that destination could be tried. The probability for that route to still be valid should be rather high. Another is that AODV only supports the use of symmetric links. When a node receives a RREQ, it will setup a reverse route to the source by using the node that forwarded the RREQ as next hop.

4.3 Definition and Data Structure of KHPCA

KHPCA is fully distributed. In the process of algorithm implementation, nodes form clusters on-demand along with the execution of routing protocol. When the routing is finished, the establishment of clusters is also accomplished. Furthermore, the cluster structure will keep relatively stable while the nodes move.

In KHPCA, each node is given a distinct sequence number. Moreover, mobile nodes have four possible states: ***init*** (node's initiate status), ***mem*** (member nodes of cluster), ***cc*** (candidate-clusterhead in a cluster) and ***clusterhead*** (node is clusterhead in a cluster). The translation relationship of each state is shown in figure 4.2.

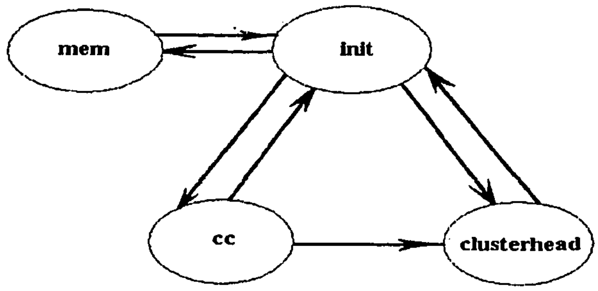


Figure 4.2: States translation in KHPCA

The definition of Candidate-Clusterhead(s) in KHPCA is as follows:

Candidate-Clusterhead(s): one or a set of several nodes within a cluster which is (are) 1-hop neighbor(s) of current clusterhead. The maximum number of hops between each candidate-clusterhead with any other nodes within one cluster is K. K is the parameter of cluster size which indicates cluster diameter.

Clusterhead is responsible for managing and maintaining the set of candidate-clusterheads. As long as current clusterhead is unavailable, the candidate-clusterhead which has the lowest node id in the set of candidate-clusterhead is going to be elected as new clusterhead.

In the sequel we use the following notations and abbreviations. The character "*i*" in these notations and abbreviations is just a variable which indicates the different nodes, clusters or clusterheads.

ID_i: the sequence number of node *i* ;

N(v): the set of nodes which are node *v*'s neighbors.

M(v): the set of member nodes. Initial value is ***nil.*** This set is be updated only if node *v* is clusterhead.

C(v): the set of candidate-clusterheads. Initiate value is ***nil.***

J(v): the set of nodes which ask for joining cluster. Initiate value is ***nil.*** This set is be updated only if node *v* is clusterhead.

N_i: the sequence number of cluster *i*. Initiate value is ***nil.*** If node *i* were clusterhead, N_i is equals to **ID_i**.

D_i: the number of node *i* 's neighbors.

C_i: the id of clusterhead *i* . Initiate value is ***nil.***

K_i: the amount of nodes in one set of candidate-clusterheads.

init status: the initiate status of any node in the network is ***init.*** Meanwhile, M(v), N(v), C(v) and J(v) are all empty. As long as source node *i* needs a path toward destination node *j* and routing is successful, the status of node *i* turns to one of other status.

mem status: nodes are member nodes (ordinary nodes) in a cluster.

cc status: nodes are candidate-clusterheads. When in this status, nodes conserve all information of member nodes within their cluster. However, they do not carry responsibilities of clusterhead and behave as ordinary nodes. Candidate-clusterhead becomes clusterhead only when needed. Consequently, it can avoid high overhead of re-clustering.

clusterhead status: node is clusterhead of a cluster. One and only one node can be clusterhead in one cluster.

To implement KHPCA, we made some modifications to RREQs of AODV. In AODV, A RREQ contains the following fields:

<source_addr, source_sequence_#, broadcast_id, dest_addr, dest_sequence_#, hop_cnt>.

We added two fields into RREQs. One is a pre-defined cluster size parameter---"K", i.e. the maximum hops within one cluster. The other is the sequence number of cluster---"N_i". Cluster structure is established while RREQs broadcast the network.

4.4 Algorithm Description of KHPCA

Generally, KHPCA is based on passive clustering philosophy. We implement it by extending AODV routing protocol. Furthermore, KHPCA is relatively independent to AODV, since clusters are established in the process of AODV routing. Therefore, the following algorithm description only aims at the establishment of cluster structure, and does not include the formation of AODV routing.

Starting at **init** status, source node which originates routing initiates cluster id N_i and sets up itself as clusterhead. Cluster id equals node sequence number, i.e. $N_i = ID_i$. Then, it broadcasts RREQ packets to its neighbor nodes. RREQs will collect the relative information of passing nodes and update their corresponding fields in packets.

In addition, modulus operation, a common mathematical method, is used to determine the maximum hops within one cluster (cluster diameter). That is, we utilize the remainder of the hops between source node and receiving node divided by parameter K to define the cluster formation.

Step 1: When a node receives the RREQ at the first time, it checks RREQ and :

i. Electing itself as the candidate-clusterhead if it is 1-hop neighbor of clusterhead. And then, it updates the RREQ and rebroadcast RREQ to its neighbors;

ii. Joining the current cluster N_i only if the remainder of hops of RREQ package between source node and this node mod K is less than K and does not equal to 1(node is't clusterhead's 1-hop neighbor), meanwhile, it does not belong to any cluster. Afterwards, it updates the RREQ and rebroadcast RREQ to its neighbors;

iii. Setting up a new cluster if the remainder of hops of RREQ package between source node and this node mod K equals to 1,meanwhile, it does not belong to any cluster and it is not 1-hop neighbor of clusterhead. At the same time, it designates cluster id as its sequence number (node id), i.e. $N_i = ID_i$. Afterwards, it updates the RREQ and rebroadcast RREQ to its neighbors. Repeating step 1 until RREQs broadcast all nodes in network.

Step 2: Nodes discard RREQ package if they receive additional copies of the same RREQ;

Step 3: Clusterhead periodically synchronizes routing information to the candidate-clusterheads within its cluster. As long as current clusterhead is unavailable, the candidate-clusterhead which has the lowest node id is going to be elected as new clusterhead.

The algorithm can be described in the following pseudocode. (figure 4.3)

We give an example of cluster formation using KHPCA. Figure 4.4 is an original network topology in an ad hoc network. We define parameter K as 2, that is, the maximum hops (cluster diameter) between any nodes within one cluster is 2. First of all, node 1 initiates routing and elects itself as clusterhead. So that Cluster id is 1. Then, node 1 sends RREQ to its neighbor nodes which is node 2 in this example. When node 2 receives RREQ and it is node 1's neighbor, it sets up itself as a candidate-clusterhead. Then, node 2 sends RREQ to its neighbor nodes which are node 3 and node 4. Consequently, node 3 and node 4 join in cluster 1 because the hops of RREQ from source node 1 to node 3 or 4 mod K (2 in this example) is zero. Whereafter, node 3 and node 4 relay RREQ to their neighbor nodes, node 11 and node 19 respectively. Node 11 forms a new cluster and elects itself as clusterhead, since it does not belong to any cluster before receiving RREQ, also, the hops of RREQ from source node 1 to it mod K is 1. Subsequently, node 11 rebroadcast RREQ to its neighbors. Similarly, node 19 forms another new cluster and elects itself as clusterhead. Any node which receives RREQ more than once ignores additional copies of RREQ packages. Eventually, clustered structure is formed as routing is accomplished.

```
for each node running KHPCA{
if aodv initiate

{
Ni=IDi    // set up itself as clusterhead
For each chi in N(ID)
Send(RREQs,chi) // send RREQs to their neighbors
}
if received(RREQs){
if(Ni in N(ID) and Ni not in M(ID)){   // node is neighbor of clusterhead
Node.state = cc
Ci = Ni
C(Ci) = C(Ci) + Ni    // join in set of candidate-clusterheads
For each chi in N(ID)
Send(RREQs,chi)
}
if(RREQs.hops mod A <>1 and Ni not in M(ID) ){
Node.state = mem
M(Ni) = C(Ni) + Ni
For each chi in N(ID)
Send(RREQs,chi)
}
else if(RREQs.hops mod A =1 and Ni not in M(ID) ){
      Node.state = clusterhead
      Ni=IDi
      For each chi in N(ID)
      Send(RREQs,chi)
}
}}
```

Figure 4.3: KHPCA algorithm

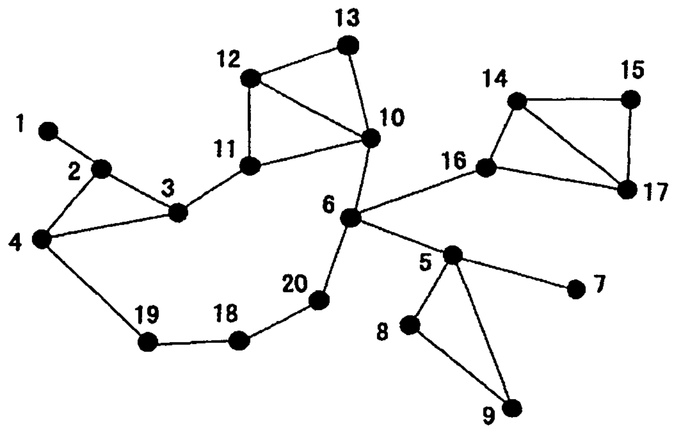


Figure 4.4: Example of network topology before using clustering algorithm

Consequently, network is clustered by five clusters. The clustered network structure after executing KHPCA is shown in figure 4.5. Five clusters are {1,2,3,4}, {11,12,13,10,6}, {19,18,20}, {16,14,15,17} and {5,7,8,9}. The clusterheads in each cluster are {1}, {11}, {19}, {16}, and {5} respectively. The set of clusterheads are {2}, {12,10}, {18}, {14,17}, and {7,8} accordingly.

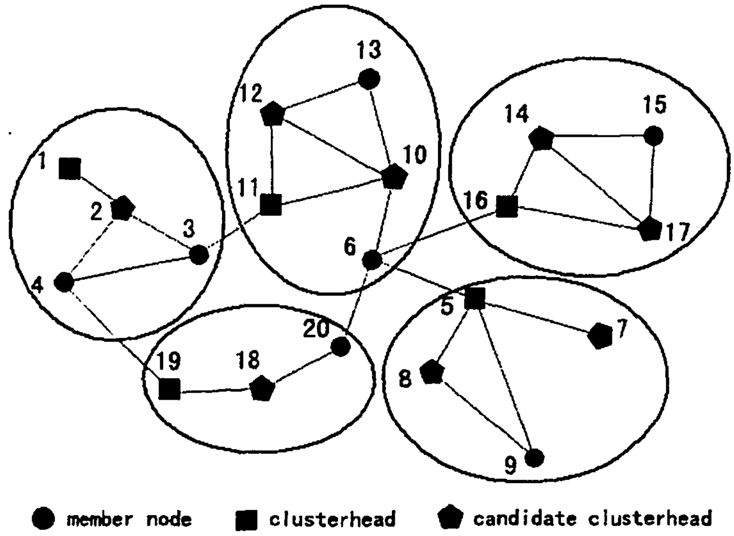


Figure 4.5: Cluster formation using KHPCA

Cluster infrastructure is maintained as follows. For instance, in cluster {11,12,13,10,6}, if clusterhead, 11, is invalid, node 10 becomes the new clusterhead because it has lowest id among candidate-clusterheads. Consequently, node 12, 13 and 6 become candidate-clusterheads. Cluster id keeps fixed to ensure continuity of clusters.

4.5 Mechanism Analysis of KHPCA

According to above algorithm description, we can easily find out that KHPCA has following characteristics.

1. KHPCA is simple. The implementation of our proposed clustering algorithm, KHPCA, utilizes the direct modification of RREQ message package of AODV. Therefore, formation of cluster structure can be accomplished simultaneously along with routing process. This algorithm does not require nodes to broadcast neighbor information repeatedly. Consequently, there is no overhead in collecting neighbor information. Furthermore, clustering can be achieved without using periodical explicit control packets or signals.
2. KHPCA is stable, and it would not cause re-clustering frequently. Through the pre-defined cluster size parameter K, users are allowed to prescribe their own K according to realistic network scope. For example, in large ad hoc networks or where nodes are intensely distributed, users can enlarge cluster size properly by increasing K. With this approach, clustering avoids excessive small independent clusters which would cause frequent re-clustering and large transmission overheads while nodes move randomly.

On the other hand, in small or particular ad hoc networks (e.g. bluetooth), user can designate K as 2, i.e. transmission range between any cluster members is equal to or less than 2 hops. In this way, KHPCA behaves similarly to the normal passive clustering mechanism and ensures reuse of limited network resource.

3. KHPCA converges very fast. Using the neighbor information carried by data packets, the cluster infrastructure can be constructed in the period of network routing. Algorithm stops when routing is accomplished.
4. KHPCA possesses fault tolerance. According to unique characteristic of passive clustering, clustering can be done without collecting complete neighbor information. If nodes move or network errors happen, KHPCA still can construct cluster infrastructure along with AODV routing.

To prove the correctness of the algorithm we have to show that: 1) every node eventually determines its cluster; 2) in a cluster, any two nodes are at most K hops away.

Lemma 1: Every node can determine its cluster and only one cluster in KHPCA.

Proof: KHPCA is executed through AODV routing process. Since at each hop RREQ broadcasts to entire network scenario, nodes can determine their clusters. The broadcasting of RREQ stops when AODV routing finishes. Therefore, each node in the network can determine its cluster and only one cluster.

Lemma 2: In a cluster, any two nodes are at most K hops away in KHPCA.

Proof: Along with each hop of RREQ broadcasting, every receiving node calculates the remainder of hops from source node to it divided by K. If hops mod K is equals to 1, node constructs another new cluster, or it will be either candidate-clusterhead or member nodes of its cluster. At least, any two nodes within its cluster can be connected by this node. Therefore, any two nodes within one cluster are at most K hops away.

Theorem 1: The algorithm terminates.

Proof: Since every node can determine its cluster *(Lemma 1)*, the algorithm will terminate when the routing is accomplished.

Theorem 2: The time complexity of KHPCA is O (|V|). (V is the set of nodes)

Proof: According to KHPCA, clustering only needs source node to broadcast RREQ packet. Thus, each node only transmits one message during the clustering, i.e. only one message is processed in system at one time. There are only |V| messages in the system. Therefore, the time complexity of KHPCA is O(|V|).

4.6 Introduction of Simulation Platform

In this thesis, we use a scalable simulation environment called GloMoSim (Global Mobile Information System Simulator) to implement KHPCA [1],[4],[49],[53],[56],[57]. Developed at UCLA (University of California, Los Angels), GloMoSim is a *library-based* sequential and parallel simulator for wireless networks. It is designed as a set of library modules, each of which simulates a specific wireless communication protocol in the protocol stack.

The library has been built using the PARSEC, a C-based parallel simulation language [3]. Designed for sequential and parallel execution of discrete-event simulation models, PARSEC can also be used as a parallel programming language. It runs on several platforms, including most recent UNIX variants as well as Windows. One of the important distinguishing features of PARSEC is its ability to execute a discrete-event simulation model using several different asynchronous parallel simulation protocols on a variety of parallel architectures. PARSEC is designed to cleanly separate the description of a simulation model from the underlying simulation protocol, sequential or parallel, used to execute it. Thus, with few modifications, a PARSEC program may be executed using the traditional sequential (Global Event List) simulation protocol or one of many parallel optimistic or conservative protocols. In addition, PARSEC provides powerful message receiving constructs that result in shorter and more natural simulation programs. New protocols and modules can be programmed and added to the library using this language.

GloMoSim is designed to be extensible and composable. It has been implemented on both shared memory and distributed memory computers and can be executed using a variety of synchronization protocols. As most network systems adopt a layered architecture, GloMoSim is being designed using a layered approach similar to the OSI seven layer network architecture. Figure 4.6 shows the GloMoSim architecture. Simple APIs are defined between different simulation layers. This allows the rapid integration of models developed at different layers by different people. Actual operational code can also be easily integrated into GloMoSim with this layered design, which is ideal for a simulation model as it has already been validated in real life and no abstraction is introduced. For example, a TCP model was implemented in GloMoSim by extracting actual code from the FreeBSD operating system. This also reduces the amount of coding required to develop the model. A number of protocols have been developed at each layer and models of these protocols or layers can be developed at different levels of granularity. Figure 4.7 lists the GloMoSim models currently available at each of the major layers. For instance, the channel propagation layer includes a *free space* model that calculates signal strength based only on the distance between every source and receiver pair.

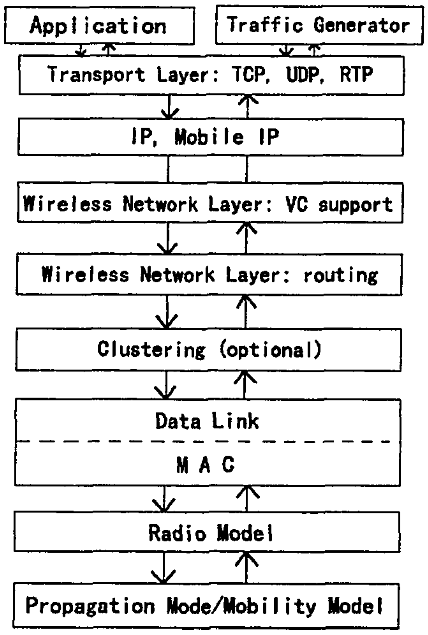


Figure 4.6: GloMoSim architecture

Layers	Models
Physical (Radio propagation)	Free space, Rayleigh, Ricean, SIRCIM
Data Link (MAC)	CSMA, MACA, MACAW, FAMA, 802.11
Network (Routing)	Flooding, Bellman-Ford, OSPF, DSR, WRP
Transport	TCP, UDP
Application	Telnet, FTP

Figure 4.7: Models in the GloMoSim library

In contrast to existing network simulators such as OPNET and NS, GloMoSim has been designed and built with the primary goal of simulating very large network models that can scale up to a million nodes using parallel simulation to significantly reduce execution times of the simulation model.

4.7 Simulation Studies

We simulate KHPCA using the GloMoSim library. Our primary aims of simulation study are to investigate the network stability, robustness and fault tolerance capacity using KHPCA. To prove our conclusion, the Lowest-ID clustering algorithm (LIDCP), a classic clustering mechanism, is used as a counterpart in our simulation. To ensure the fairness and precision of simulation results, we execute every following simulating experiment 20 times and choose the average values for reference. The random-way point model is used for node mobility [8]. In the first series of simulations, 100 nodes are distributed randomly in a 2000m × 2000m space. Radio propagation range for each node is 150 meters. We designate parameter K as 2 in this small network environment. Mobile nodes have identical transmission power and move at the speed from 0 to 10m/s in any direction freely. In addition, nodes initiate routing within 200 seconds randomly.

To measure network stability, we investigate the change of clusters within a time unit of both clustering schemes. Figure 4.8 uses changes of cluster member nodes which are caused by the changes of clusters to represent the changes of clusters in the network.

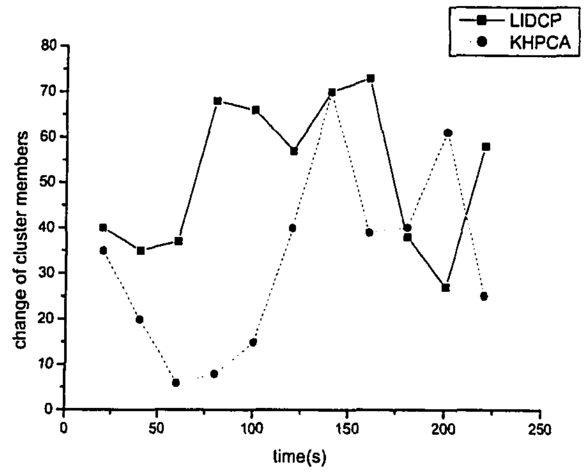


Figure 4.8 Change of cluster member nodes

This figure indicates that, with KHPCA, the changes of clusters are significantly less than the changes using Lowest-ID clustering. The result just proved that passive clustering could solve "chain reaction" problem much better than conventional clustering algorithms. Compared with the Lowest-ID clustering algorithm, better network stability can be achieved using KHPCA.

Figure 4.9 illustrates clusters survival time using different clustering algorithms. As time passes, the number of clusters using KHPCA fluctuates between 12 to 24, the number of clusters using node id based non-clusterhead clustering algorithm fluctuates between 26 to 34. Meanwhile, the number of clusters using Lowest-ID fluctuates between 24 to 50 drastically as time passes. Obviously, KHPCA has better performance in clusters survival time than Lowest-ID but worse than non-clusterhead approach. That is because non-clusterhead clustering does not consider cluster disappearance caused by the change of clusterheads. On the other hand, Lowest-ID causes much more frequent re-clustering as the change of clusterheads. In KHPCA, if current clusterhead moves out of its cluster, the candidate-clusterhead which has lowest node id would act as new clusterhead and maintains stability of current cluster. Therefore, the clustered infrastructure using KHPCA is more stable than other clusterhead-based clustering schemes, and the network is more robust.

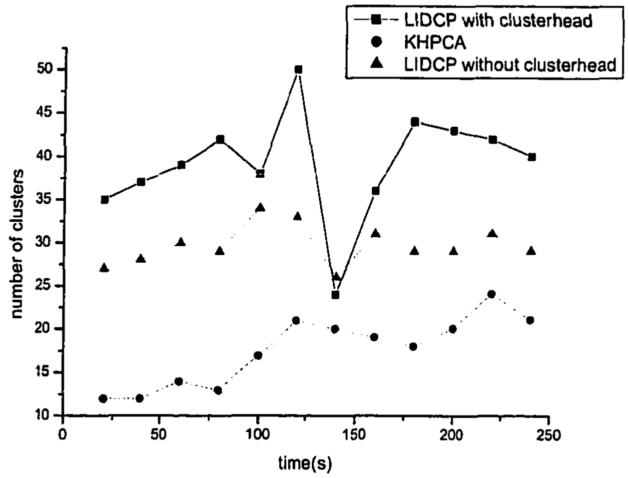


Figure 4.9: Clusters survival time

Figure 4.10 indicates the change of cluster numbers along with increased fault nodes. Similarly, KHPCA performs better in fault tolerance than Lowest-ID clustering. Even though a large number of nodes are invalid, KHPCA can maintain relatively stable cluster infrastructure, since clusters would not vanish immediately as clusterheads change.

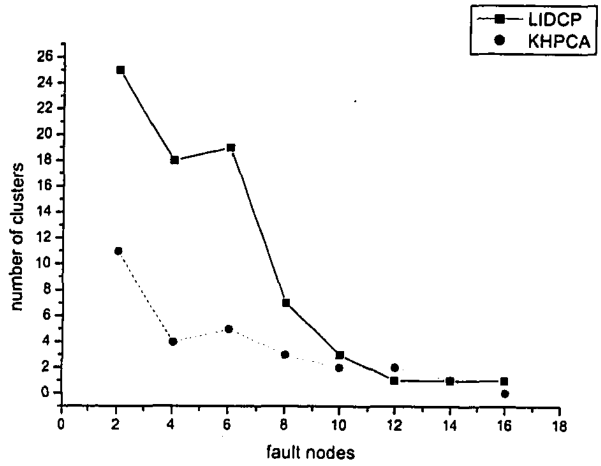


Figure 4.10: fault tolerance capacity

In the next series of simulations, we enlarge the network scope approximately two times. Meanwhile, we increase the value of parameter K to 3. That is, we expand the cluster size in KHPCA. Compared with Lowest-ID and KHPCA with K = 2 (2HPCA) in this larger network environment, we investigate the characteristics of KHPCA with K = 3 (3HPCA) in network stability, robustness and fault tolerance capacity. In this simulation environment, 500 mobile nodes are distributed randomly in a 3000m × 3000m space. The other network assumptions and conditions keep unaltered with the first series of simulations.

Figure 4.11 indicates that, when using 2HPCA and 3HPCA, both of the cluster changes are much less than the cluster changes using Lowest-ID clustering mechanism because they solve "chain reaction" much better than conventional clustering algorithms. Furthermore, 3HPCA performs even better than 2HPCA which performs similarly to common passive clustering algorithm, since the larger cluster size and more candidate-clusterheads can avoid frequent changes of clusters and network stability is more preferable.

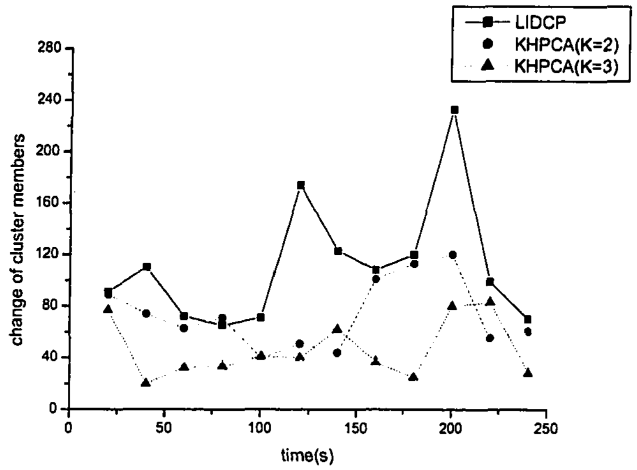


Figure 4.11 Change of cluster member nodes (K=3)

From figure 4.12 we can easily find out that the number of clusters using Lowest-ID clustering fluctuates drastically within the wide range from 46 to 132. That is because Lowest-ID clustering produces excessive small clusters and then causes frequent re-clustering while clusters change to satisfy the criteria of clusterhead election. 3HPCA performs best in this simulation. Analytically, larger clusters would respond less sensitively than small ones when clusterheads change. Moreover, the plentiful candidate-clusterheads which can be elected as new clusterheads instantly ensure the relatively stable clustered infrastructure and lower probability of re-clustering. Therefore, the network is robust when using 3HPCA.

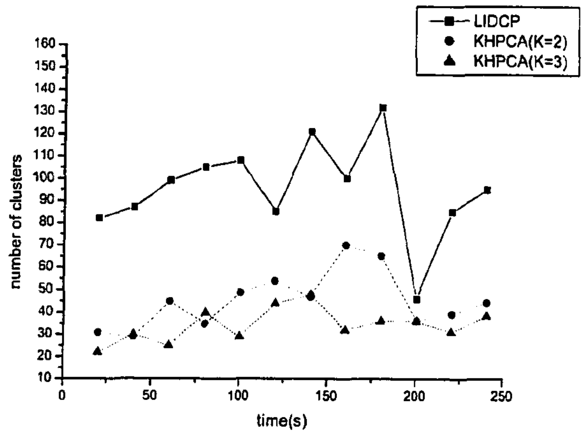


Figure 4.12: Clusters survival time (K=3)

The following figure 4.13 indicates that 3HPCA performs better in fault tolerance than Lowest-ID clustering and 2HPCA, because clusters would not disappear immediately as clusterheads change.

Summarily, simulation results show that the network is more stable, robust and has higher fault tolerance capacity when using KHPCA. In small or particular ad hoc networks, we could designate cluster size parameter K as 2 to ensure the reuse of limited network resource. On the other hand, in large ad hoc networks or where mobile nodes are densely distributed, we could simply increase parameter K not only for avoiding excessive small clusters and frequent re-clustering, but also for achieving stable and robust network.

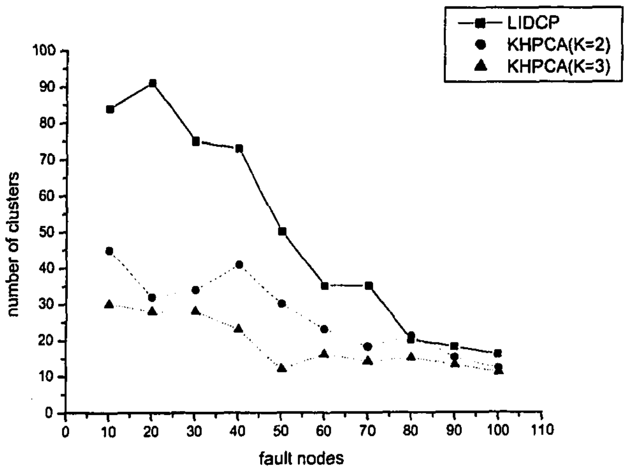


Figure 4.13: fault tolerance capacity (K=3)

CHAPTER 5

Conclusion

5.1 Conclusion

With the rapid development of wireless technology, the applications of wireless networks are blooming. As a significant wireless infrastructure, Ad hoc networking has attracted more and more attention by network experts. Leaning on extraordinary characteristics, the development and applications of ad hoc networking will play a leading role in future wireless communications. This thesis focuses on clustering mechanisms in a hierarchical ad hoc network architecture.

This thesis summarizes the development of wireless technology and the most important issues in ad hoc networks. Furthermore, this thesis concentrates in current clustering algorithms in ad hoc networks. First of all, we analyzed existing representative clustering algorithms and classified them into seven categories. Then, identifying limitations in previous clustering schemes, we have presented a new clustering algorithm called KHPCA (K-Hops Passive Clustering Algorithm) for ad hoc networks. We proposed the concept of "candidate-clusterheads" which is beneficial for network stability and reliability. New clusterheads can be simply obtained from the set of candidate-clusterheads but not from frequent re-clustering. KHPCA is simple, stable and has better fault tolerance capacity than conventional clustering algorithms. Finally, we implemented our new proposed algorithm in GloMoSim library. Comparison results between KHPCA and classic Lowest-ID show the correctness of KHPCA. Furthermore, KHPCA achieved our design goal and its merits indicate that it has good research value.

5.2 Open Issues

1. In KHPCA, we simply elect 1-hop neighbors of clusterheads as candidate-clusterheads. In the future, with the availability of particular network scenario and wireless hardware, we may consider more realistic network parameters in selecting candidate-clusterheads, such as transmission capacity, power constraint of each node and specific node movement. By integrating these parameters, we may select candidate-clusterheads and new clusterhead more fairly and reasonably. Consequently, we could maintain optimum cluster infrastructure within a fixed time period.
2. According to the unique characteristics of AODV, we may reduce the overheads of control information but maintain cluster stability. In addition, the ultimate goal is that we can achieve best performance in clustered ad hoc networks by ideal combination of the clustering mechanism and routing. However, this goal only can be done in a future more mature test bed.

Bibliography:

[1]. H. Akhtar, "An Overview of Some Network Modeling, Simulation, &Performance Analysis Tools", Proceedings of 2nd IEEE Symposium on Computers and Communications, 1997.

[2]. A. D. Amis, R. Prakash, D. Huynh, and T. Vuong, "Max-min D-cluster formation in wireless ad hoc networks," in IEEE INFOCOM, 2000, pp. 32–41.

[3]. R.Bagrodia, R. Meyerr, et al., "PARSEC: A Parallel Simulation Environment for Complex System", UCLA technical report, 1997.

[4]. L. Bajaj, M. Takai, R. Ahuja, K. Tang, R. Bagrodia, and M. Gerla, "GlomoSim: A Scalable Network Simulation Environment", Technical Report CSD Technical Report, #990027, UCLA, 1999.

[5]. S. Basagni, "Distributed clustering for ad hoc networks," in International Symposium on Parallel Architectures, Algorithms, and Networks, 1999.

[6]. S. Basagni, "Finding a Maximal Weighted Independent Set inWireless Networks", Telecommunication Systems 18:1–3, 155–168, 2001.

[7]. Christian Bettstetter and Stefan Konig, "On the Message and Time Complexity of a Distributed Mobility-Adaptive Clustering Algorithm in Wireless Ad Hoc Networks",TUM, 2001.

[8]. J. Broch et al, "A performance comparison of multi-hop wireless ad hoc network routing protocols", In ACM Mobicom'98, Oct. 1998.

[9]. J.Broch, D.B.Johson, and D.A.Maltz, "Dynamic Source Routing Protocol for Mobile Ad Hoc Wireless Networks", IETF Internet draft, October 1999.

[10]. Ian D. Chakeres and Elizabeth M. Belding-Royer, "AODV Routing Protocol Implementation Design", Proceedings of the International Workshop on Wireless Ad Hoc Networking (WWAN), Tokyo, Japan, March 2004.

[11]. M. Chatterjee, S. K. Das, and D. Turgut, "WCA: A weighted clustering algorithm for mobile ad hoc networks," Journal of Cluster Computing (Special Issue on Mobile Ad hoc Networks), vol. 5, no. 2, pp. 193–204, April 2002.

[12]. C.C.Chiang, "Routing in Clustered Multihop, Mobile Wireless Network with Fading Channel", In Proceedings of IEEE SICON'97, April 1997, pp.197-211.

[13]. Carlos de Morais Cordeiro and Dharma P. Agrawal, "Mobile Ad Hoc Networking", OBR Research Center for Distributed and Mobile Computing,ECECS, University of Cincinnati,2001.

[14]. R.Dube, C.D.Rais, K-Y.Wang,and S.K.Tripathi, "Singal Stability-Based Adaptive Routing (SSA) for Ad-Hoc Mobile Networks", IEEE Personal Communications, February 1997, pp.36-45.

[15]. A. Ephremides, J. E. Wieselthier, and D. J. Baker, "A design concept for reliable mobile radio networks with frequency hopping signaling", Proc. of IEEE, 75(1):56-73, Jan. 1987.

[16]. L.R.Ford, Jr., and D.R.Fulkerson, "Flows in Networks", Princeton University Press,Princeton, NJ, 1962.

[17]. Magnus Frodigh, Per Johansson and Peter Larsson, "Wireless ad hoc networking-The art of networking without a network", Ericsson Review No. 4, 2000.

[18]. B. Garbinato and Rupp, P., "From Ad Hoc Networks to Ad Hoc Applications", Proceedings of the 7th International Conference on Telecommunications, 2003.

[19]. M. Gerla, Taek Jin Kwon, G. Pei, "On-demand routing in large ad hoc wireless networks with passive clustering", Wireless Communications and Networking Conference, 2000. WCNC. 2000 IEEE, 23-28 Sept. 2000 pp.100-105 vol.1.

[20]. Z. J. Haas, "The routing algorithm for the reconfigurable wireless networks," in Proceedings of IEEE International Conference on Universal Personal Communications (ICUPC'97), San Diego, California, Oct. 1997.

[21]. T. C. Hou, T sai T J, "Adaptive clustering in a hierarchical Ad Hoc networks", Proceedings of International Computer Symposium'98, 1998,pp. 171-176.

[22]. D.B.Johson, "Routing in Ad Hoc Networks of Mobile hosts", Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications, December 1994, pp.158-163.

[23]. J.Jubin and J.D.Tornow, "The DARPA packet radio network protocol", Proc. Of the IEEE, vol 75, No.1, Jan 1987, pp.21-32.

[24]. H. Knospe, Schwiderski-Grosche S, "Future mobile networks: ad-hoc access based on online payment with smartcards". Personal,Indoor and Mobile Radio Communications, 2002. The 13th IEEE International Symposium on, 15-18 Sept. 2002 pp.197-200 vol.1.

[25]. P. Krishna, N. H. Vaidya, M. Chatterjee, and D. K. Pradhan, "A cluster-based approach for routing in dynamic networks," in SIGCOMM Computer Communications Review (CCR), 1997.

[26]. Taek Jin Kwon, Mario Gerla, "Efficient Flooding in Ad hoc Networks using On-Demand (Passive) Cluster Formation", MOBIHOC 2002.

[27]. T. J. Kwon and M. Gerla, "Clustering with power control," in IEEE MILCOM, 1999.

[28]. Taek Jin Kwon, Gerla M, Varma V K, et al, "Efficient flooding with passive clustering an over-heard-free selective forward mechanism for ad hoc/sensor networks", Proceedings of the IEEE; Vol.91 Issue: 8, Aug. 2003, pp.1210-1220.

[29]. J. A. Laurila, J. Mikkonen, Rinnemaa J,"Wireless LAN access network architecture for mobile operators".Communications Magazine, IEEE, Vol.39 Issue: 11, Nov. 2001 pp.82-89.

[30]. B.M.Leiner, D.L.Nielson, and F.A.Tobagi, "Issues in Packet Radio Network Design", Proceedings of IEEE (Special Issue, Packet Radio Networks), Vol.75, No.1, January 1987, pp.6-20.

[31]. C. R. Lin and M. Gerla, "Adaptive clustering for mobile wireless networks," IEEE Journal on Selected Areas in Communications, vol. 15, pp. 1265–1275, September 1997.

[32]. C. R. Lin and M. Gerla, "Adaptive Clustering for Mobile Wireless Networks", IEEE Wireless Networks, 1995,1(3) pp.255-265.

[33]. C. R. Lin and M. Gerla, "A distributed control scheme in multi-hop packet radio networks for voice/data traffic support", in: Proceedings of IEEE GLOBECOM (1995) pp. 1238–1242.

[34]. C. R. Lin and M. Gerla, "A distributed architecture for multimedia in dynamic wireless networks", in: Proceedings of IEEE GLOBECOM (1995) pp. 1468–1472.

[35]. A.B.McDonald and T.F.Znati, "A mobility-based framework for adaptive clustering in wireless ad hoc networks', IEEE Journal on Selected Areas in Communications,17(8):1466-1486,Aug. 1999.

[36]. S.Murthy and J.J.Garcia-Luna-Aceves, " An Efficient Routing Protocol for Wireless Networks", In Proceedings of ACM Mobile Networks and Applications Journal, Special Issue on Routing in Mobile Communication Networks, October 1996, pp.183-197.

[37]. M. Joa-Ng and I.-T. Lu, "A GPS-based peer-to-peer hierarchical link state routing for mobile ad hoc networks," in 51st IEEE Vehicular Technology Conference, 2000.

[38]. M. Joa-Ng and I.-T. Lu, "A peer-to-peer zone-based two-level link state routing for mobile ad hoc networks", IEEE Journal on Selected Areas in Communications (August 1999) 1415–1425.

[39]. Abhay K. Parekh, "Selecting Routers in Ad-Hoc Wireless Networks", In *ITS*, 1994.

[40]. V.D.Park and M.S.Corson, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks", In Proceedings of INFOCOM'97, April 1997.

[41]. C.E.Perkins, "Ad Hoc Networking", Addison-Wesley,December 2000.

[42]. C.E.Perkins and P.Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers", In Proceedings of the conference on Communications architectures, protocols and applications, London, England, August 31-September 2, 1994, pp.234-244.

[43]. C.E.Perkins and M.E.Royer, "Ad-hoc On-Demand Distance Vector Routing", In Proceedings of the Second Annual IEEE Workshop on Mobile Computing Systems and Applications, February 1999.

[44]. C.E.Perkins, M.E.Royer and S.R.Das, "Ad-hoc On-Demand Distance Vector Routing", IETF Internet draft, July 2000.

[45]. Elizabeth M. Royer, "Routing approaches in mobile ad hoc networks", in: S. Basagni, M. Conti, S. Giordano,I. Stojmenovic (Eds.), Ad Hoc Networking, IEEE Press Wiley, New York, 2003.

[46]. Elizabeth M. Royer, "Multi-Level Hierarchies for Scalable Ad hoc Routing", University of California, Santa Barbara, CA 93106,2002.

[47]. Elizabeth M. Royer and Charles E. Perkins, "An Implementation Study of the AODV Routing Protocol", Proceedings of the IEEE Wireless Communications and Networking Conference, Chicago, IL, September 2000.

[48]. Elizabeth M. Royer, C.-K. Toh, "A review of current routing protocols for ad-hoc mobile wireless networks", IEEE Personal Communications Magazine (April 1999) 46–55.

[49]. Yoav Sasson, David Cavin and Andre Schiper, "On the Accuracy of MANET Simulators", POMC'02, October 30-31, 2002.

[50]. C.K.Toh, "Associativity-Based Routing for Ad-Hoc Networks", Wireless Personal Communications, vol.4, No.2, March 1997, pp.1-36.

[51]. J. T. Tsai and M. Gerla, "Multicluster, mobile, multimedia radio network," ACM/Kluwer Journal of Wireless Networks, vol. 1, no. 3, pp. 255–265, 1995.

[52]. Damla Turgut, Sajal K. Das, Ramez Elmasri and Begumhan Turgut, "Optimizing Clustering Algorithm in Mobile Ad hoc Networks Using Genetic Algorithmic Approach",2002.

[53]. Xiang Zeng, Rajive Bagrodia, Mario Gerla, "GloMoSim: A Library for Parallel Simulation of Large-scale Wireless Networks", Workshop on Parallel and Distributed Simulation, 1998 pp 154-161.

[54]. An Zhu, Jie Gao, Leonidas J.Guibas, John Hershberger, Li Zhang, "Discrete Mobile Centers", Discrete and Computational Geometry, pages 45-63, Volume 30, Number 1, May 2003.

[55]. Mobile ad hoc Networks (MANET). URL: http://www.ietf.org/html.charters/manetcharter .html

[56]. The GloMoSim installation guide: http://www.home.mylkcn.net/gaoanyu/glomosim.html

[57]. The official GloMoSim website: http://pcl.cs.ucla.edu/projects/glomosim/