2004

Infrastructureless wireless networks: cluster-based architectures and protocols

Jamal Nazzal Al-Karaki
Iowa State University

Recommended Citation

Al-Karaki, Jamal Nazzal, "Infrastructureless wireless networks: cluster-based architectures and protocols " (2004). *Retrospective Theses and Dissertations*. 762.
https://lib.dr.iastate.edu/rtd/762

Infrastructureless wireless networks: Cluster-based architectures and protocols

by

Jamal Nazzal Al-Karaki

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Computer Engineering

Program of Study Committee:
Ahmed E. Kamal, Major Professor
Arun K. Somani
Douglas Jacobson
Manimaran Govindarasu
Sigurdur Olafsson

Iowa State University

Ames, Iowa

2004

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

Graduate College
Iowa State University

This is to certify that the doctoral dissertation of

Jamal Nazzal Al-Karaki

has met the dissertation requirements of Iowa State University

Signature was redacted for privacy.

Major Professor

Signature was redacted for privacy.

For the Major Program

DEDICATION

To my **Mother** and **Father** for their loving guidance and sacrifice ...
To my wife **Maram** and our children **Sharaf** and **Jenah** ...
and
to my big **Family** for their endless love and support.

ACKNOWLEDGEMENTS

My utmost thanks should go to ALLAH (swt) for His guidance and love for me throughout my life. I am indebted to many individuals for their care and support given to me during my doctoral studies. First of all, I would like to express my deep gratitude to Professor Ahmed E. Kamal. As my thesis advisor, he has provided me constant encouragement, insightful comments, and invaluable suggestions, which benefited not only the completion of this thesis, but also my career in a long time to come. I have learned enormously from him not just how to do research, but also how to communicate it effectively to others. I was so lucky to be part of his group since he is so concerned about his students. Prof. Kamal's influence is strong in this work. He critically read drafts of all chapters and the appendices, pointing out errors ranging from grammatical slips to technical blunders. He clarified many obscurities, and freely gave me the benefit of his insight. I am deeply grateful.

I would also like to thank Prof. Arun K. Somani for his valuable feedback and suggestions on this dissertation. His breadth of knowledge and his enthusiasm for research always amaze me. I am so grateful to Dr. Manimaran Govindarasu for his initial feedback and guidance at the initial stages of my graduate career, especially in the general area of multimedia networking and for reading and commenting on my thesis. His suggestions were very helpful. There are many good teachers at ISU, but I was fortunate enough to work with one outstanding teacher, Dr. Sigurdur Olafsson. I thank him for showing me what it takes to be a great teacher and for all his advice and for reading and commenting on my thesis. I also thank him for many useful conversations we had while serving as my mentor in the Preparing Future Faculty (PFF) Program at Iowa State. I am so thankful to Dr. Doug Jacobson for serving on my dissertation committee and for his inspirational teaching style. His feedback about my dissertation was valuable.

I would like to take this opportunity to express my thanks to those who helped me with various aspects of conducting research and the writing of this thesis while at ISU. I am also grateful to the former and present members of the Laboratory for Advanced Networks (LAN). I had the privilege of interacting with wonderful, bright, and talented people. They have taught me much, and their advice, feedback, and friendship have made my Ph.D. experience both more educational and more fun. In particular, I acknowledge Raza Ul-Mustafa, Basheer Duwairi, Mohammad Fraiwan, Ashraf Hamad, Haider Qulaibu, Saad Safiullah, Qiang Qiu, Samyukta Sankaran, Murli Viswanathan, Ahmed Younis,

Ra'ed Al-Omari, Mezyad Ammourah, Saqib Malik, Mohammed Khailfeh, Anirban Chakrabarti, and all the friends in Ames. I am really pleased to be part of all these talented colleagues. I hope to continue my close ties with them in the future. I would especially like to thank Raza Ul-Mustafa for all his help with data aggregation algorithms—I've enjoyed our collaborations! Raza sincerity and vision always inspire me. Thanks to Mohammad Fraiwan for teaching me some MATLAB tricks. Thanks also go to my sponsor, the Hashemite University, and to the ECpE Department of ISU for providing financial support during my Ph.D. program.

I would like to thank my family, my sisters, and my brothers for all their love and support. I am very lucky to have such wonderful family members. I am indebted to my parents for everything that they have given to me. They taught me the value of knowledge, the joy of love, and the importance of family. They have stood by me in everything I have done, providing constant support, encouragement, and love. They are an inspiration to me in all that they do. My special thanks should go to my mother since she always have loved me, believed in me, and encouraged me to succeed in this life. She taught me that giving is much more important than taking especially in knowledge.

Finally, I would like to thank my wonderful wife, Maram. Maram has been my true companion since the day I met her almost 3 years ago. She has kept me happy and sane during the Ph.D. process, and I thank her for all her patience and her never-ending optimism when I came home late, frustrated, and stressed. Without her dedicated sacrifice, support, understanding, and encouragement, this dissertation could not be completed. If I forgot someone, I cannot forget the ones who keep my life filled with laughter and love, and I know that there is nothing I cannot accomplish with the smiles of my children Sharaf and Jenah always inspiring me.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

An important challenge in the design of wireless and mobile systems is that two key resources — communication bandwidth and energy — are significantly more limited than in a tethered network environment. In addition, the time-varying characteristics of wireless channels make it hard to consistently obtain the same performance. These restrictions require innovative communication, networking, and design techniques for the efficient utilization of the bandwidth and energy. One of the most rapidly developing areas in wireless networks is wireless ad hoc networks. A wireless ad hoc network is an autonomous system consisting of nodes, which may or may not be mobile, connected with wireless links and without using pre-existing communication infrastructure or central control. Ad hoc networking is expected to play an important role in future wireless mobile networks due to the the widespread use of mobile and hand-held devices. Mobile Ad hoc Networks (MANETs) and Wireless Sensor Networks (WSNs) are two prominent classes of these *infrastructureless* wireless networks. While MANETs exhibit dynamic topology changes due to free node mobility, WSNs have unreplinishible energy limitations. Hence, topology control, Quality of Service (QoS) routing, and power control become challenging issues. We argue that cluster-based techniques coupled with cross-layer design can achieve better performance in this harsh environment. This dissertation supports this claim by introducing strategies for topology control, QoS routing in MANETs, and energy efficient routing in WSNs. First, we develop the Virtual Grid Architecture (VGA), which is a fixed and stable architecture for ad hoc networks that can support efficient routing and network control. We show that although VGA clustering is simple, it is close to optimal. Then, we develop two QoS routing protocols that combine the ideas of cluster-based routing and cross layer design to achieve good performance in terms of delay, bandwidth, and user-perceived quality. These protocols, operating on top of VGA, show improved system call success rate and packet delivery ratio by an order of magnitude compared to general-purpose approaches. Finally, we develop GRASP (Grid-based Routing and Aggregator Selection Protocols), a scheme for WSNs, that combines the ideas of fixed cluster-based routing of VGA together with application-specific data aggregation functions. GRASP is able to enhance the network performance in terms of extending the network lifetime, while incurring acceptable levels of latency in data aggregation. Our studies together show that creating stable and scalable architecture can achieve topology robustness, enhance quality of service, and attain the energy and latency efficiency needed for wireless networks.

CHAPTER 1 Introduction

Wireless networks have experienced unprecedented development in the past decade. In fact, it is predicted that wireless data access will exceed wireline access by the current year 2004 (see Figure 1.1) [1]. The past several years have shown a wealth of new protocols for wireless networks, including both routing and Medium Access Control (MAC) protocols. However, the convergence of wireless technologies including cellular phones, pagers, laptops, and personal digital assistants (PDAs), presents many new challenges in order to make the *anytime*, *anywhere* computing paradigm real and effective. Ad hoc networks take this concept one step further by envisioning networks without any fixed infrastructure or central control.

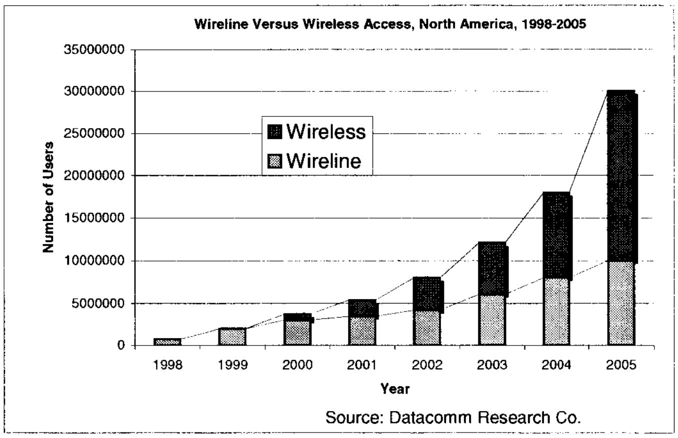


Figure 1.1 Predictions are that wireless data access will exceed wireline access by the year 2004 as reported by Datacomm Research Co. [1].

Ad hoc networks are a new paradigm of networks that form dynamically by some autonomous mobile nodes that are connected together through wireless links, and without using any fixed infrastructure or centralized control. Nodes maintain connectivity in a decentralized manner. Moreover, if the nodes are mobile, the network topology may change in a rapid and unpredictable manner due to node mobility. This causes frequent link outage, which may cause route breakage and service disruptions. As such, routing, network connectivity, and network stability become challenging issues. Indeed, a careful observation of the human behavior indicates that human society itself can be seen as a real life example of ad hoc wireless networks.

1.1 Infrastructerless Wireless Networks: Background

Advances in energy-efficient design and wireless technologies have enabled portable devices to support several important wireless applications, including real-time multimedia communication, surveillance using sensor networks, and home networking applications [4]. An important challenge in the design of wireless and mobile systems is that two key resources — communication bandwidth and energy — are significantly more limited than in a tethered network environment. These restrictions require innovative communication techniques to increase the amount of bandwidth per user and innovative design techniques and protocols to use available energy efficiently. Furthermore, wireless channels are inherently error-prone and their time-varying characteristics make it hard to consistently obtain good performance. Therefore, communication protocols must be designed to always try to adapt to current conditions instead of being designed for worst-case conditions.

One of the most rapidly developing areas in wireless networks is ad hoc wireless networks (also called multihop wireless networks). Ad hoc networking is expected to play an important role in future wireless mobile networks [2]. The widespread use of mobile and handheld devices is likely to popularize wireless ad hoc networks. Conventional wireless networks (e.g., cellular networks or satellite networks) rely on a fixed infrastructure, e.g., fixed base-stations and wired communications, which connects the users always by routing data through these fixed base-stations. Thus, traditional wireless networks are usually called *Infrastructured* wireless networks. Installing such an infrastructure is often either too expensive or technically impossible for some remote localities. In contrast to conventional wireless networks, the most distinctive feature of ad hoc wireless networks is the lack of any fixed or preexisting network infrastructure, and hence they are referred to as *Infrastructerless* wireless networks. Ad hoc wireless networks mitigate the lack of infrastructure by allowing users to route data through intermediate nodes, where each mobile node can act as a relay to forward traffic toward the destination (hop-by-hop routing). Figure 1.2 shows an example of both (a) Infrastructered wireless networks where mobile nodes are connected by a wired backbone, and (b) Infrastructerless wireless networks, where all communications are using the wireless medium.

Physically, a wireless ad hoc network, or simply an ad hoc network, consists of a number of geographically distributed, potentially mobile, nodes sharing a common radio channel. Ad hoc networks are self-creating, self-organizing, and self configuring. They are self-creating networks because when nodes get together, an ad hoc network is created *on the fly* as the nodes communicate with each other, i.e., ad hoc networks come into being solely by interactions among their constituent wireless mobile nodes. Ad hoc networks are self-organizing because only such interactions are used to provide the necessary control and administration functions supporting such networks. Moreover, nodes cooperate to organize

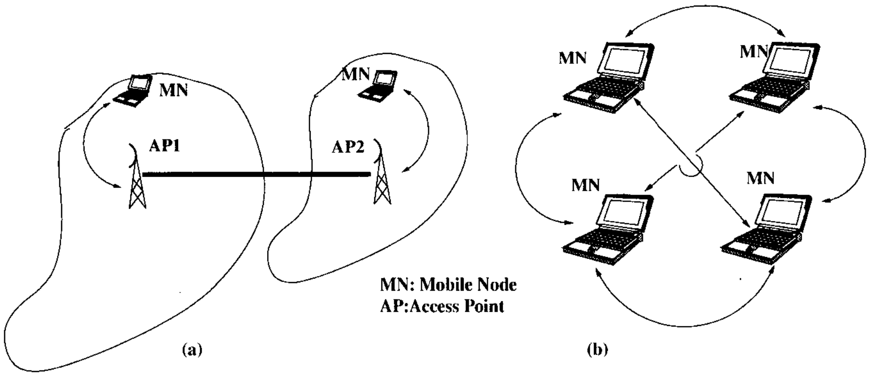


Figure 1.2 The difference between (a) Infrastructered wireless networks, and (b) Infrastructerless wireless networks.

themselves and distribute different roles if necessary. Finally, these networks are self configuring since the network does not depend on a particular node as a central controller and dynamically adjusts as nodes join or leave the network due to free node mobility. As such, networks like these are both flexible and robust.

Due to their flexibility and robustness, ad hoc networks can be quickly deployed for the support of many applications. For example, ad hoc networks are very useful in military and other tactical applications such as battlefield, search and rescue, and disaster relief. While the military is still a major driving force behind the development of these networks, ad hoc networks are quickly finding new applications in civilian areas. Ad hoc networks will enable people to exchange data in the field or in classroom without using any network structure except the one they create by simply turning on their computers or PDAs, or enable a flock of sensors to form a self-organizing group and collectively perform some monitoring task. Commercial applications are also likely where there is a need for ubiquitous communication services without the presence or use of a fixed infrastructure, e.g., spontaneous communication between mobile computers for conferencing and home networking, multihop extensions of cellular telecommunication systems, and networks of vehicles. As wireless communication increasingly permeates everyday life, new applications for mobile ad hoc networks will continue to emerge and become an important part of the communication structure. The intrinsic features of Infrastructerless Wireless Networks (IWNs), which distinguish them from other networks, can be summarized as follows:

- **Lack of fixed infrastructure:** Ad hoc networks are designed to work without the need for existing infrastructure. Hence, every node acts in a distributed peer-to-peer fashion, without relying on any centralized infrastructure for control and administration. However, a virtual backbone can be constructed by selecting a subset of the mobile nodes in the network to act as virtual base-stations. The set of selected nodes can be used to perform routing and other network management functions.

- **Dynamic topology:** In IWNs, nodes are allowed to move and in random fashion. Hence, network topology may change rapidly and unpredictably. This can cause frequent link outages and packets losses, and may also lead to frequent network partitions. Hence, the design of routing protocols is a crucial and challenging problem in IWNs. This problem is even more exacerbated when paths need to satisfy certain quality of service guarantees during the connection lifetime. The network state kept by a node is always a best approximation of the current network state, whose precision degrades as the network topology changes.

- **Multi-hop routing:** In IWNs, every node has to behave as a router in order to relay messages for other nodes. This motivates the need for mechanisms that stimulate or enforce nodes to cooperate especially for environments where nodes tend to behave selfishly. The multihop topology of IWNs may also allow for spatial reuse of the wireless spectrum, where two nodes can transmit using the same bandwidth, provided they are sufficiently apart.

- **Node heterogeneity and link variability:** IWNs are typically heterogeneous networks with various types of mobile nodes forming together an ad hoc network. For example, different military units ranging from soldiers to tanks or mobile devices ranging from sensors to laptops can come together to form an IWN. Thus, mobile nodes will have different packet generation rates, routing responsibilities, network activities, and power source capacities. The node heterogeneity can affect the design and performance of communication protocols in IWNs. The problem of variable link quality is particularly significant in IWNs. This affects the main Quality of Service (QoS) parameters such as bandwidth availability, latency, reliability, and jitter. It is preferable that scheduling algorithms or communication protocol designed for IWNs to be channel state aware in order to avoid bad links when possible.

- **Scarce resources:** Almost all nodes of ad hoc networks are battery powered. Hence, innovative techniques to eliminate energy inefficiencies that would shorten the lifetime of the nodes are required. An effective method to increase the capacity of a wireless network is *power control.* Although power control has been traditionally studied at the physical layer, it impacts every aspect of the network protocol stack. Moreover, bandwidth in wireless networks is scarcer than in wireline networks. Consequently, the wireless spectrum is a limited resource which must be utilized efficiently. Since bandwidth availability has direct effect on the QoS, effective management of this resource is a key factor for supporting QoS in IWNs.

- **Limited physical security:** Due to the lack of central control and the characteristics of wireless communications, ad hoc networks are more exposed to security threats than wireline networks. This motivates the need for security mechanisms that suite the nature of these networks. In particular, any security protocol should be completely distributed and does not assume any central entity in control.

Despite of the various difficult design issues inherent in these networks, mobile computers and applications will become indispensable even at times when and at places where the necessary infrastructure is not available. In concept, wireless computing devices should physically be able to communicate with each other, even when no routers or base-stations, or Internet Service Providers (ISPs) can be found. In the absence of infrastructure, what is needed is that the wireless devices themselves take on the missing functions. This is the main theme of wireless ad hoc networks. However, for this vision to become a reality, many research challenges pertaining to these networks have to be resolved. In the following section, we discuss many challenges that are open for research in IWNs.

1.2 Research Challenges in Infrastructerless Wireless Networks

Along with the numerous advantages of ad hoc networks come challenges in design and implementation. In particular, the free node mobility, the decentralized nature of the network, and the scarce resources impose significant research challenges in all layers of the protocol stack. Furthermore, these networks inherit all the problems of traditional wireless networking, e.g., undesirable time varying properties of wireless channel and some side effects such as the hidden-terminal and exposed-terminal problems.

Given the features of IWNs, the design of practical protocols for ad hoc networks is often driven by many conflicting factors. On one hand, many problems in these networks are inherently difficult (NP-complete). As a result, many researchers were forced to look for suboptimal solutions. On the other hand, the high costs, in terms of the computation and communication overhead, associated with many efficient algorithms that were originally designed for wireline networks, limit their practical usages in the wireless environment. Hence, designing communication protocols for ad hoc networks is an interesting, but difficult topic. To resolve these conflicts, new algorithms and solutions for all layers of the protocol stack will be needed.

Although IWNs comprise many classes (e.g., Rooftop [6], Bluetooth [7], HomeRF [8], etc.), the two mostly researched classes of IWNs are: Mobile Ad hoc networks (MANETs) and Wireless Sensor networks (WSNs), and they have some common as well as unique characteristics as described later in this chapter. Among the various aspects of these networks, topology control, routing, quality of service, and energy-efficient communication are the most active research areas. In the following, we summarize some of the research challenges in both MANETs and WSNs.

1.2.1 Mobile Ad hoc Networks (MANETs)

The concept of MANETs dates back to the DARPA packet radio network program in the 1970's [5]. The renewed interest in these networks in recent years is largely due to the recent developments in mobile computing and wireless technologies. Among other aspects, MANETs are distinguished from other IWNs classes by the feature of dynamic topology, which is mainly attributed to free node movements. Because the network topology changes arbitrarily as the nodes move, routing information is subject to becoming obsolete, and different nodes often have different views of the network, both in time (information may be outdated at some nodes but current at others) and in space (a node may only know the network topology in its neighborhood and not far away from itself). The ephemeral node associations in MANETs limit the link lifetime, thus affecting the route lifetime. As such, the most challenging issue in MANETs is routing, which is further exacerbated when routes need to satisfy certain quality of service (QoS) guarantees in terms of bandwidth or end-to-end delays. Figure 1.3 shows an example of a MANET where mobile nodes are using the wireless channel to make peer-to-peer communication through single or multihop paths. Since gathering fresh information about the entire network is often both costly and impractical, many routing protocols in MANETs are on-demand protocols, i.e., they collect routing information only when necessary, and to destinations they need routes to. By doing this, routing overhead is greatly reduced when compared to the traditional *proactive* protocols, which maintain optimal routes to all destinations at all time. A comprehensive survey of traditional routing protocols in MANETs can be found in [9].

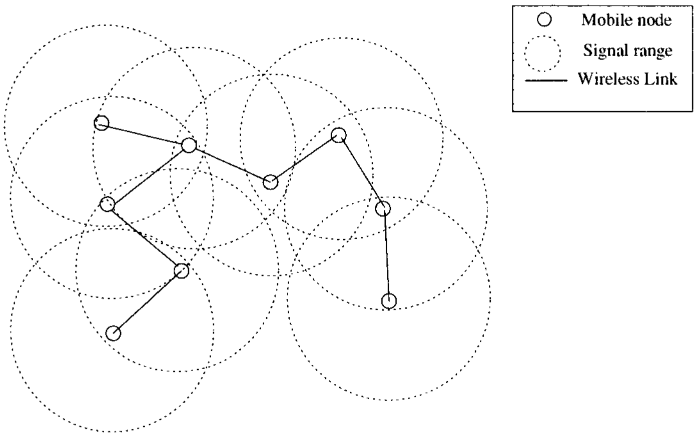


Figure 1.3 An example of multihop connectivity in Mobile Ad hoc Networks (MANETs).

In addition to routing, many research challenges in MANETs require efforts from researchers in order for these networks to become common place. In particular, the following research topics are very important in the design of future MANETs:

- **Topology Control:** Topology management and control is an active area of research in MANETs.

An alternative for physical infrastructure is the construction of a virtual backbone or infrastructure. A virtual backbone plays a very important role in routing, where the number of nodes responsible for routing can be reduced to the number of nodes in the backbone. The virtual backbone also plays an important role for data broadcasting and connectivity management in wireless ad hoc networks. Efficient topology control algorithms are needed for MANETs.

- **Quality-of-service (QoS) routing:** Routing is the most actively researched area in MANETs. Routing becomes challenging when the route has to satisfy certain QoS guarantees, e.g., bandwidth, end-to-end delay, and packet loss ratio. Most of the previously proposed routing protocols in MANETs address the issue of routing from a single-layer perspective, that is, at the network layer. Recently, it has been concluded that inter-layer dependencies play a critical role in providing an efficient and comprehensive solution to the QoS routing problem [12], and this view is a key design principle that we capitalize on in the QoS routing protocols proposed in this dissertation. The use of cross-layer design optimization has been successfully demonstrated in the context of wireless Internet delivery and protocol frameworks for active wireless networks [3]. The impact of layers interaction design (interoperability) and the effect of this on the network performance has not been documented yet in a quantitative way. Indeed, layer interdependencies are more pronounced in MANETs.

1.2.2 Wireless Sensor Networks (WSNs)

Advances in sensor technology have enabled the development of small, relatively inexpensive and low-power sensors. A sensor is any device that maps a physical quantity from the environment to a quantitative measurement. Sensor nodes are equipped with a sensor module (e.g., acoustic, seismic, image, etc.) capable of sensing some quantity about the environment, a digital processor for processing the signals from the sensor and performing network protocol functions, a radio module for communication, and a battery to provide energy for operation (see Figure 1.4). The position of the sensor nodes need not be engineered or predetermined and hence allows random deployment in inaccessible terrain or disaster relief operations. This implies that the nodes are expected to perform sensing and communication with no continual maintenance or human attendance.

Wireless Sensor Networks (WSNs) are a class of IWNs that contain hundreds or thousands of sensor nodes. Each node has the ability to sense elements of its environment, perform simple computations, and communicate either among its peers or directly to an external base-station (BS), and hence allowing for monitoring and control of various physical parameters [10]. Future WSNs are envisioned to revolutionize the paradigm of collecting and processing information in diverse environments. However, the severe energy constraints and limited computing resources of the sensors present major challenges for such a vision to become a reality. WSNs have limited lifetimes since sensors are powered by lim-

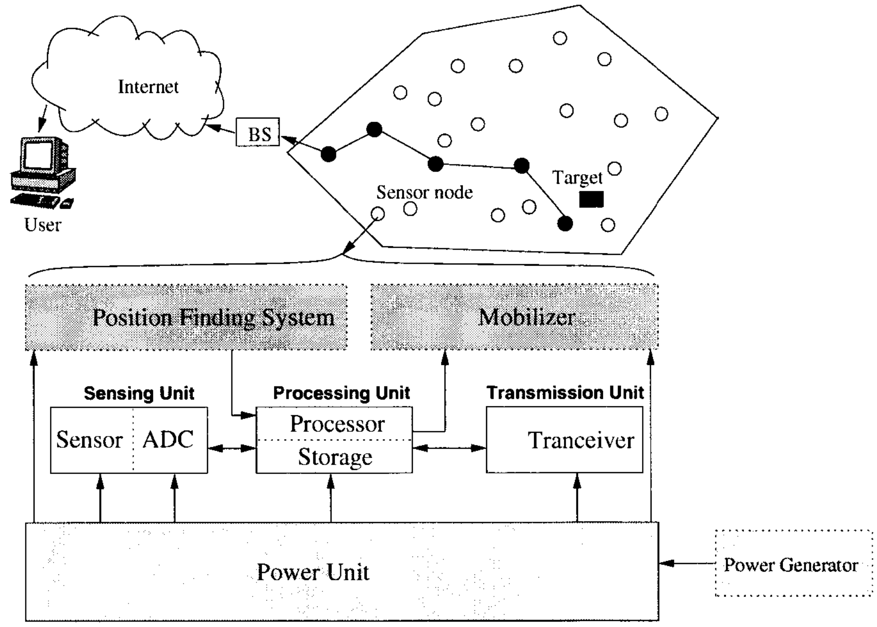


Figure 1.4 Wireless Sensor network structure and sensor node components.

ited energy batteries. Thus, innovative techniques to eliminate energy inefficiencies that would shorten the lifetime of sensor nodes are needed. Moreover, sensor nodes have limited computing power and cannot run sophisticated network protocols. Therefore, light-weight communication protocols are favorable in WSNs. Some application examples of WSNs are target field imaging, intrusion detection, weather monitoring, security and tactical surveillance, distributed computing, detecting ambient conditions such as temperature, movement, sound, light, or the presence of certain objects, and inventory control. Figure 1.5 shows a sensor network used in a battlefield of a military application.

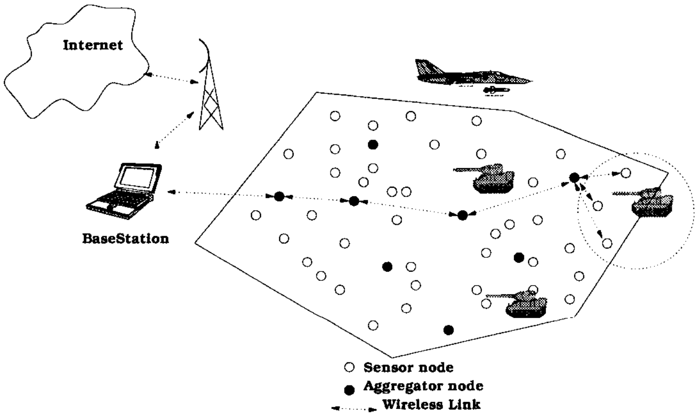


Figure 1.5 An example of wireless sensor network (WSN) used for remote monitoring of battle field.

The intrinsic features of Wireless sensor networks can be summarized as follows:

1. Sensor networks are application specific (i.e. design requirements of a sensor network change with application). For example, the challenging problem of low-latency precision tactical surveillance is different from that required for a periodic weather-monitoring task.

2. In many situations, data at neighboring sensor nodes is not independent since sensor nodes are monitoring a common phenomena, and hence their reported data can be redundant or highly correlated (data redundancy). Therefore, similar data can be first combined or aggregated and then sent to destinations in order to reduce number of transmissions and allow users to accurately and reliably monitor an environment. As such, significant savings in energy usage can be obtained.

3. Sensor networks are *data-centric* networks. In traditional networks, data is normally requested from a specific node. In sensor networks, data is requested based on certain attribute-value pairs. For example, if the the following query is (temperature > 60F) is sent to a WSN, then sensor nodes that sense temperature > 60F only need to respond and report their readings. The sensors can remain in the sleep state, with the data being reported from the few remaining sensors providing lower quality. Once an event of interest is detected, the system can respond and configure itself so as to obtain very high quality results.

4. Position awareness of sensor nodes is important since data collection is based on the location. With the current technology, it is not feasible to use Global Positioning System (GPS) hardware for this purpose. Other GPS-free methods are needed.

Both MANETs and WSNs share the same research problems, including: the time varying characteristics of wireless links, the multi-hop wireless communication, the limited power supply, the possibility of link failures, and the ad hoc deployment of nodes in the network area. However, WSNs differ from MANETs in several ways. First, the destination in WSNs is known (e.g. external base-station (BS)). Communication in WSNs is many-to-one, while in MANETs it is generally on a peer-to-peer basis. Second, data collected by many sensors in WSNs is based on common phenomena, hence there is a high probability that this data has some redundancy. Third, MANETs are characterized by highly dynamic topologies due to free node mobility. Table 1.1 shows a high level comparison between MANETs and WSNs by highlighting the main differences.

Despite the innumerable applications of WSNs, these networks have several restrictions, e.g., limited energy supply, limited computing power, and limited bandwidth of the wireless links connecting sensor nodes. Hence, the effective use and deployment of WSNs still faces several problems, and these problems need to be addressed and resolved. In the following, we summarize some of those problems and some of the design factors that can be considered:

1. **Energy Consumption without losing Accuracy:** Sensor nodes can use up their limited supply of energy performing computations and transmitting information in a wireless environment.

Table 1.1 MANETs versus WSNs: A high level Comparison.

MANETs	WSNs
Dynamic Topologies	Static Topologies (in most cases)
MANETs are typically heterogeneous networks (Tens-hundreds of nodes)	WSNs are typically homogeneous networks (thousands and more)
Power can be renewed, or recharged	Need to operate for long time on a tiny battery
Destination is unknown	Destination in WSNs is known
Address-centric routing	Data-Centric routing
Use point-to-point communications	Use many-to-one communication
Optimize QoS and performance	Optimize Power use not quality.

As such, energy-conserving forms of communication and computation are essential as sensor node lifetime shows a strong dependence on battery lifetime. One of the main design goals of WSNs is to prolong the lifetime of the network and prevent connectivity degradation by employing aggressive energy management techniques.

2. **Fault tolerance:** Some sensor nodes may fail or become blocked due to lack of power, physical damage, or environmental interference. The failure of sensor nodes should not affect the overall task of the sensor network. If many nodes fail, MAC and routing protocols must accommodate formation of new links and routes to the data collection base stations. Therefore, multiple levels of redundancy may be needed to have a fault-tolerant sensor network.

3. **Scalability:** The number of sensor nodes deployed in the sensing area may be in the order of hundreds or thousands, or more. Any scheme must be able to work with this huge number of sensor nodes. Also, change in network size, node density, and topology should not affect the task and operation of the sensor network.

4. **Transmission media:** In general, the required bandwidth of sensor data will be low, on the order of 1-100 kb/s. Related to the transmission media is the design of medium access control (MAC). Appropriate protocols are needed that are tailored to the specific demands of WSNs.

5. **Quality of Service:** In some applications, data sensed or events occurring in the environment can be time-sensitive, and must be communicated in a timely manner. Long delays due to processing or communication may be unacceptable or render data useless. Therefore, it is often important to bound the end-to-end latency of data dissemination.

6. **Connectivity:** Node density in sensor networks precludes them from being completely isolated from each other. Therefore, sensor nodes are expected to be highly connected. This, however,

may not prevent the network topology from being variable due to node failures. Hence, ensuring network connectivity is a basic requirement for communication in these networks.

7. **Coverage:** Each sensor node obtains a certain *view* of the environment. A given sensor's view of the environment is limited both in range and in accuracy; it can only cover a limited physical area of the environment. Hence, area coverage is also an important design parameter in WSNs.

To summarize, WSNs protocols should be (a) self-configuring, in order to enable ease of deployment of the networks, (b) energy-efficient and robust, to extend system lifetime, and (c) latency-aware, to get the information to the enduser as quickly as possible in some critical applications. The research on WSNs presented in this dissertation focuses on ways in which features (a), (b), and (c) are considered when designing protocol architecture in these networks.

1.3 Contributions of this Dissertation

In this dissertation, we identify several prominent problems of IWNs and present several state-of-the-art solutions with a focus on energy efficiency and quality of service issues. The work in this dissertation is based on the premise that new innovative protocols and architectures must be designed and deployed in order to achieve high performance and energy-efficient wireless ad hoc environment. The problems of topology control, transmission scheduling, power control, quality-of-service routing, and energy efficient routing in IWNs are investigated in this dissertation. We realize that most of the routing problems in ad hoc networks, whether they relate to QoS routing in MANETs or energy-efficient routing in WSNs, are partly due to the unstable topology in MANETs and severe energy-constrained nodes in WSNs in addition to the random deployment of nodes in the network area. The free mobility of a large number of mobile nodes in ad hoc networks can have profound effects on routing, including network state book-keeping, energy consumption, etc. Therefore we introduce a fixed, stable, and connected virtual wireless backbone that is robust against frequent link failures in MANETs and can support energy-efficient communication in WSNs. Such a stable backbone can substantially help minimize the effect of the above issues. We investigate providing such a stable topology with low overhead. Furthermore, conventional network architectures and protocols are designed according to a layering approach where each layer of the system is designed separately and is independent of the application. We argue that ad hoc networks will be more efficient if they are designed to exploit interactions among layers and specific features of the applications they are supporting. We show that by using cluster-based approach coupled with cross-layer design, our protocols and architectures can achieve the high performance and energy efficiency needed under the tight constraints of a tetherless environment.

Three key problems are investigated for MANETs: (1) the need to maintain a minimum quality of service over time-varying channels, (2) to operate with limited energy resources, and (3) to operate in a heterogeneous environment with variable topologies. We identify two main principles to solve these problems. The first principle is that, in order to provide QoS, an essential topology control mechanisms that result in a stable virtual topology are required. In particular, having a fixed, stable, and connected routing architecture (i.e., virtual wireless backbone) that is robust against frequent link failures in MANETs and can support energy-efficient communication in WSNs can substantially simplify most of the routing problems. Second, energy efficiency and QoS should benefit from layers interaction in IWNs. Due to the dynamic topology, energy limitations, and error-prone wireless environment, adaptability of the system will be a key issue in achieving these solutions.

The next step after designing the stable topology is to design protocols that can utilize the unique features of such virtual infrastructure in a manner that significantly enhance the performance of ad hoc networks. We present innovative protocols and architectures that can utilize this stable virtual architecture coupled with the cross-layer design paradigm in order to help perform routing with QoS guarantees in MANETs. The virtual topology is also used as the basis for energy-efficient data-centric based routing scheme in WSNs.

The contribution of this work can be summarized as follows.

- We address the issue of topology control in both homogeneous and heterogeneous ad hoc networks by developing a simple, fixed, and scalable virtual wireless backbone. The fixed backbone is created through a novel and simple tessellation (zoning) scheme with low overhead. The tessellation scheme maps the network physical topology onto a virtual fixed grid (*rectilinear*) topology. A simple power control scheme, which captures differences in node transmission power, is used in conjunction with the zoning process to ensure network connectivity. The virtual wireless backbone, called Virtual Grid Architecture (VGA), simplifies the routing function in infrastructerless wireless networks. VGA consists of a few, but possibly more powerful, nodes that are optimally selected and known as *clusterheads* (CHs). We study the performance of VGA both analytically and through simulation. For homogeneous networks, with a large number of users, we derive expressions for the number of clusterheads (clusterheads cardinality), VGA worst case path length, and VGA average case path length. We show mathematically why rectilinear routing is better than diagonal routing over VGA. For small- to medium-sized heterogeneous networks, we develop an Integer Linear Program (ILP) that finds the optimal number of connected clusterheads in an arbitrary connected MANETs. We also derive expressions for the communication overhead over VGA. Both analytical and simulation results show that our proposed algorithm pertaining to VGA, though being simple, is close to optimal. To show that efficient routing algorithms can be performed on top of this virtual architecture, we propose two routing strategies: one is a modified on-demand routing technique and the other is based on the concept of transitive closure. Both algorithms show improved routing performance over VGA.

- A critical requirement of ad hoc networks is that each node in the network has a *path* to every other node in the network, i.e., the network should be connected. We consider the problem of determining the connectivity of ad hoc networks under VGA clustering. In other words, we are interested in finding the probability that each CH node will have at least one path to any other CH node over VGA. Another important property, which is very useful for QoS routing, is topology stability. This later property has long been overlooked in the literature. We present an analytical treatment of the stability and connectivity features of VGA, which are fundamental to the support of QoS in MANETs. We show that the stability feature of VGA results in high packet delivery and call acceptance ratios. Furthermore, we show that VGA stability helps in performing routing with statistical end-to-end QoS guarantees, e.g., delay and bandwidth.

- We study the challenging issue of QoS routing in MANETs. We develop two QoS routing protocols for heterogeneous MANETs. The first protocol, called Virtual Grid Architecture Protocol (VGAP), can provide statistical end-to-end delay and bandwidth guarantees. VGAP operates on top of VGA, where CHs discover multiple QoS routes using an extended version of the OSPF routing protocol, called Mobile OSPF (M-OSPF). M-OSPF interacts with an extended version of WFQ scheduling policy, called Ad hoc WFQ (AWFQ) to discover Qos routes. AWFQ is a channel-state dependent scheduler that takes into account the varying time characteristics of the wireless channel, yet providing an upper bound on the statistical end-to-end delay guarantees of a certain flow. Besides being fully distributed, VGAP requires little communication overhead. It also capitalizes on the interplay among the bottom three layers of the protocol stack (i.e., physical, MAC, and network layers) to enhance the network performance. Moreover, VGAP repairs a route when it breaks. Load balancing and route redundancy can also be achieved in VGAP. The second QoS routing protocol, called Virtual Routing with Feedback (VRF), utilizes a feedback mechanisms to discover routes with smaller end-to-end delays or routes with higher energy reserve. In particular, the status of the traffic queues at mobile nodes (called traffic density) and the remaining energy level at these nodes are used as indications of the potential delay a packet may incur using a certain route or the potential lifetime of a route as far as energy is concerned. VRF selects routes that exhibit less end-to-end delays or routes that have higher potential lifetimes among the set of discovered routes over VGA.

- Finally, we design and implement Grid-based Routing and Aggregator Selection Protocol (**GRASP**), a scheme for WSNs that combines the ideas of fixed cluster-based routing together with application-specific data aggregation to achieve good performance in terms of system lifetime and latency. This approach improves system lifetime, while incurring acceptable levels of latency in data aggregation, and hence attain the energy and latency efficiency needed for wireless sensor networks without sacrificing quality. Since data are correlated and the end-user only requires a high-level description of the events occurring in the environment, sensor nodes can collaborate locally and globally to reduce the data that need to be transmitted to the end-user. Since correlation is strongest among data signals coming from nodes that are close to each other, the use of a clustering infrastructure will allow nodes that are close to each other to share data before sending it to the BS. Hence, GRASP operates over a topology similar to VGA. We derive necessary and sufficient conditions for coverage with connectivity of VGA in WSNs. GRASP performs data aggregation and in-network processing at different levels of the cluster hierarchy. GRASP presents an exact algorithm as well as several approximate algorithms for the joint problem of routing with data aggregation in WSNs. The exact algorithm is formulated as an Integer Linear Program (ILP), which can solve for small to medium sized WSNs. For large WSNs, we present several heuristics that yield faster, albeit near-optimal, solutions.

1.4 Dissertation Organization

The rest of this dissertation is organized as follows. In Chapter 2, we present the general system model, notations, and define few important terms. We also present the details of our clustering approach, namely VGA clustering, that serves as an efficient topology control algorithm in both homogeneous and heterogeneous MANETs. Two routing techniques that use VGA are discussed in this chapter. Comprehensive analytical and simulation-based performance studies of our clustering approach are carried out in Chapter 3. In Chapter 4, we present two QoS routing protocols for MANETs. Both QoS routing protocols utilize the proposed virtual topology in order to find routes with the required QoS guarantees. In Chapter 5, we present energy-efficient routing protocols for WSNs. These protocols work on top of the virtual topology presented in Chapter 2 and they employ data aggregation and in-network processing techniques to maximize WSNs lifetime. We also derive necessary and sufficient conditions for the connectivity and coverage in WSNs under our clustering approach. For each proposed protocol in this dissertation, we present several experimental results in each designated chapter. Finally, Chapter 6 draws several conclusions and presents future work directions. Since this dissertation presents detailed studies of two different classes of wireless ad hoc networks, we survey related work in each respected chapter rather than consolidating the related work in one chapter.

CHAPTER 2 Efficient Topology Control in Wireless Ad hoc Networks

Although Infrastructreless Wireless Networks (IWNs) have no physical backbone or infrastructure, an alternative for physical backbone is the construction of a virtual backbone. A virtual backbone plays a very important role in routing, where the number of nodes responsible for routing can be reduced to the number of nodes in the backbone. The virtual backbone also plays an important role for data broadcasting and connectivity management in IWNs. We realize that most of the routing problems in MANETs are partly due to the disconnected or unstable topology caused by random node deployment and free node movement. Hence, an important task in IWNs is to determine an appropriate virtual topology over which high-level routing protocols are implemented. In particular, there is a legitimate need for a fixed, stable, and connected routing architecture (i.e., virtual wireless backbone) that is robust against node movement and frequent link failures in MANETs. Such an architecture can simplify many of the routing problems in MANETs. In this chapter, we investigate providing such a stable topology with low overhead.

2.1 Related Work

The issue of *topology control* is very important in MANETs. One way to create infrastructure in ad hoc networks is to perform node clustering. Compared to conventional routing protocols, the cluster-based routing approach incurs lower overhead during topology state updates and also has faster convergence [42]. The reduction in state information is particularly useful in dynamic topology networks like MANETs. In the *cluster-based* philosophy, ad hoc networks are structured as a two-level or multi-level networks: in the lower level, nodes in geographical proximity create peer-to-peer networks or local clusters. In each local cluster in the lower-level network, at least one node is designated to serve as a *gateway* or *clusterhead* to the higher tier(s). In fact, these gateway nodes create the higher-level network. Routing between nodes that belong to different lower-level networks is performed through the clusterhead nodes. In other words, nodes that belong to the same local cluster send data directly from one node to another, whereas inter-cluster data are routed through the clusterhead nodes. The clusterhead nodes change either periodically, randomly, or on-demand in response to clusterhead node mobility or if dramatic changes happen in the clusterhead resources status in order to keep the network fully connected. This also has the advantage of avoiding *hot-spots*, or bottlenecks in the network. A cluster maintenance algorithm can be used to ensure connectivity of all nodes in the presence of node mobility or node heterogeneity. In fact, clustering can enable bandwidth reuse by reducing interference, and thus can increase system capacity [13].

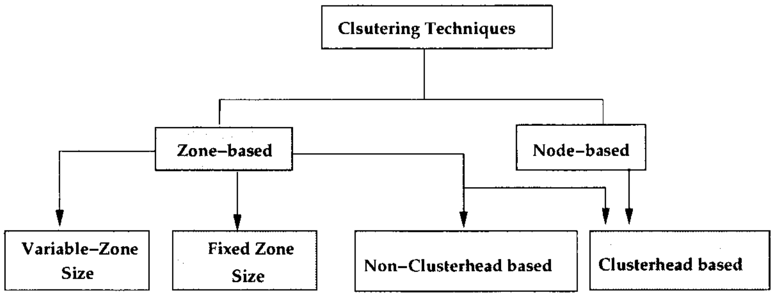


Figure 2.1 Clustering techniques in MANETs: Classification Map

Clustering techniques can be classified as *node-based* or *zone-based* clustering (see Figure 2.1 for a classification map). In the former, Clusterhead (CH) nodes are first selected and then clusters are formed by the CH nodes. In the latter, zones or clusters are first created in the network and then inside each zone a CH node is selected. In both cases, the set of clusterheads (CHs) form a virtual wireless backbone. Moreover, clustering techniques can form a variable cluster size or fixed cluster size clusters. Variable size clustering techniques (see Figure 2.2) or dynamic virtual backbone demands complex cluster management schemes in order to update the virtual topology in response to frequent topological changes. Moreover, many large as well as many small size clusters can exist in the network. Hence, some clusters will be overloaded while others will be lightly loaded, and hence we are confronted with a load balancing problem. Therefore it is difficult to construct and maintain an effective hierarchical structure. In a fixed size clustering protocol, however, nodes are organized into fixed clusters, and these clusters remain fixed throughout the lifetime of the network (see Figure 2.3). If clusters have fixed sizes, the virtual topology updates will be rare. As long as each cluster is occupied by at least one mobile node, there is no need for virtual topology updates. Furthermore, clustering techniques can be broadly divided into *clusterhead-based* schemes or *non-clusterhead-based* schemes. In the former case, a clusterhead is selected in each cluster to implement the extra control functions. In the latter, no clusterhead is chosen and all mobile nodes are identical. In [28], it is shown that, for large ad hoc networks, a clusterhead-based scheme outperforms a non-clusterhead-based scheme in terms of routing overhead reduction.

The role of clusterhead (CH) is a temporary role, which changes dynamically as the topology or other factors affecting it change. The set of CHs may form a dominating set, which is a set of nodes such that each node is either in the set or is adjacent to a node in the set. The minimum dominating set (MDS) problem asks for a dominating set of minimum size whereas minimum connected dominating set

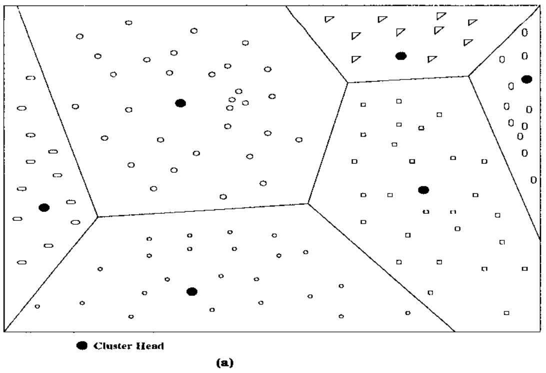


Figure 2.2 Variable-size clustering technique

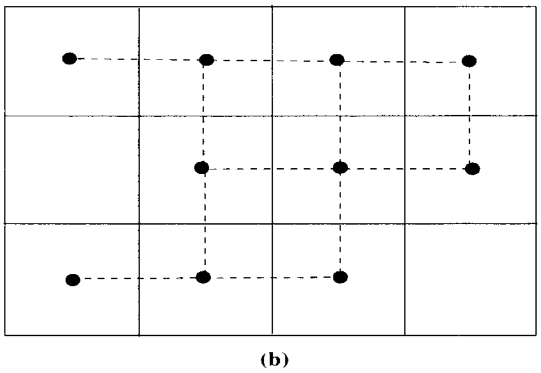


Figure 2.3 Fixed-size clustering technique.

(MCDS) is an MDS with CHs communicating with each other either directly or indirectly but through other CHs only (so called 0-hop minimum dominating set). However, finding the minimal set of CHs, or equivalently, finding the minimum connected dominating set (MCDS), is an NP-Complete problem [52]. Therefore, most approaches are based on heuristic solutions that can be far from the optimal solution. If the optimal solution is the target, then scalability becomes an issue in the sense that for large networks, the developed algorithms become computationally infeasible [45].

In fact, the concept of virtual backbones is not new. Early works on this issue appeared in [21][22][23]. However, the authors in these works do not attempt to optimize the size of the virtual backbones. Following the early efforts, clustering in MANETs received a considerable amount of research [26]-[38]. Many approximate algorithms for MDS and MCDS have been proposed in the literature. Most of these algorithms suffer from very poor approximation ratio, ranging from $O(\log n)$ to $O(n)$ and from high time complexity and message complexity, ranging from $O(n^2)$ to $O(n^3)$. Examples of clustering techniques that approximate MCDS are proposed by Das et al. [43], which has a logarithmic approximation factor and Wu and Li [32] that has linear approximation factor. Heuristics were also introduced in [43], [35], [30], [47], [29], most of which have high time complexity and/or message complexity. In [44], [34] clustering algorithms that organize the mobile nodes into non-overlapped clusters based on their IDs were proposed. In [38], a framework is suggested for dynamically organizing mobile nodes into clusters in which the probability of path availability can be bounded. In [28], a multiple access scheme is developed for control information broadcasting, based on which the Access-Based Clustering Protocol (ABCP) is proposed. The ABCP makes a clustering decision directly based on the result of channel access, which can reduce the control overhead of clustering.

In this Chapter, we propose a zone-based and fixed size clustering scheme that results in a stable virtual topology, called Virtual Grid Architecture (VGA). VGA will be used as the underlying backbone for performing simple and efficient routing in MANETs. We devise a simple and efficient zoning process that utilizes location information of mobile nodes to form *fixed* size clusters of nodes. Inside each cluster, a CH is elected and the set of these CHs form the virtual wireless backbone. The connectivity pattern of these CHs forms a *rectilinear grid*, and hence the name, Virtual Grid Architecture (VGA). Since zones are fixed, sophisticated algorithms for zone management are not needed. According to the classification map in Figure 2.1, our clustering technique is zone-based, clusterhead based, and clusters have fixed sizes. The advantages of such clustering technique will be evident as we move on in this dissertation.

Although the use of virtual wireless backbone routing in MANETs is not a new idea, our proposed clustering scheme distinguishes itself from previous schemes in several ways. First, in almost all previous schemes, variable cluster size or dynamic virtual backbone were used. Hence, complex cluster management schemes are needed to update the virtual topology in response to frequent topological changes. On the contrary, our clusters (later called zones) are fixed and the virtual topology updates are rare. This is due to the stability feature of our virtual topology and its enhanced reliability (connectivity) capabilities as shown later in this Chapter. As long as each cluster is occupied by at least one mobile node, there is no need for virtual topology updates. Second, in previous algorithms, MANETs may have many large as well as many small size clusters. Hence, some CHs will be overloaded while other CHs are lightly loaded (i.e., load balancing problem) and therefore it is difficult to construct and maintain an effective hierarchical structure. In our approach, all clusters are fixed and have the same size in homogeneous MANETs, while different zone sizes are used in heterogeneous MANETs. Thus, the resulting architecture is fixed, simple, and scalable. Third, most of the previous clustering algorithms consider the case of only homogeneous nodes where nodes are assumed identical, while our clustering approach considers both homogenous as well as the more realistic heterogeneous MANETs where the capabilities (e.g., transmission power) of mobile nodes may differ in the same network. The proposed clustering technique in this Chapter is promising since the the basic virtual topology, the rectilinear grid, is fixed. If the physical topology is to change due to the removal or reinstatement of a link, the virtual topology can easily be updated and managed accordingly.

2.2 System Model

We consider a MANET with N mobile nodes randomly distributed in a two-dimensional network area A in square meters. All nodes are equipped with batteries with possibly different levels of energy density. If nodes differ in their transmission power levels, directional links may exist between different nodes. Let the distance between two nodes i and j be d_{ij}. The condition for the existence of a link from node i to node j ($i \rightarrow j$) is given by ($d_{ij} \leq r_i$) where r_i is the transmission radius of node i. The distance is basically obtained from the radio device on each node based on the signal strength. A link is said to be bidirectional if both $i \rightarrow j$ and $j \rightarrow i$ hold. In homogeneous MANETs, nodes exhibit the same transmission power, i.e., nodes will have the same transmission range. In heterogeneous MANETs, however, mobile nodes have different characteristics and transmission ranges (e.g., laptops, PDAs, sensors, cellular nodes, etc.). As such, even when node i is located in node j's transmission range, j may not be reachable from i. This may happen when node i has shorter transmission range than node j. Mobile nodes are free to move in A, which is usually captured using a certain mobility model. We explore many mobility models including the most used ones in the literature. However, we will develop and use a more realistic mobility model that can capture node movements (see Section 2.5). Mobile nodes can find their location coordinates using a location server such as Global Positioning System (GPS), where we assume that each mobile node is equipped with a GPS card. Note that there are cases when the GPS card may not work (i.e., inside buildings) or the signal can be jammed or blocked maliciously (e.g., battle fields). In this case, a GPS-free approach [15], which uses methods based on triangulation or multilateration, can be used. Such methods allow mobile nodes to approximate their positions using radio signal strength or time of flight from a few other nodes.

We apply a fixed clustering approach in the next section that results in a stable virtual wireless backbone. The virtual backbone consists of a set of nodes called clusterheads (CHs). The set of CHs and the wireless links connecting them is modeled as a directed graph $G = (V, E)$, where V is the set of CH nodes and E is the set of directed wireless links connecting CH nodes. Each nonempty zone corresponds to a vertex in the graph G. Vertical and horizontal adjacent vertices (CHs) are connected by an edge if the distance between them is less than or equal to the corresponding transmission range and the two adjacent zones are not empty. A link is absent from E if either one or the two neighboring zones are empty or a communication in the respective direction is not possible. We formally define the QoS route as follows. A QoS route $P_{(s,d)}$ on the virtual rectilinear graph G with a length of n hops and connecting a source-destination pair (s, d) can be represented as a sequence of vertices $v_0, ..., v_i, ..., v_n$ where $0 < i < n$ and $v_0 = s, \;\; v_n = d$. Each link l, $l \in E$, has two associated link metrics, namely, b_l and d_l which correspond to the bandwidth capacity and the packet delay on link l, respectively. Each link l on the path $P_{(s,d)}$ starts at vertex v_{i-1} and ends at vertex v_i for $1 \leq i \leq n$. We consider a session-oriented traffic, where each unidirectional session is also called a flow. We assume that a session requires minimum bandwidth and maximum end-to-end delay. A source mobile node issues a call to a certain destination with specific QoS requirements in the form $\langle s, d, R_{bw}, D_{max} \rangle$ where R_{bw} is the minimum required bandwidth and D_{max} is the maximum end-to-end delay. A call is admitted if R_{bw} and D_{max} can be satisfied, i.e.,

$$R_{bw} \leq \min_{l \in P_{(s,d)}} b_l \quad ; \; D_{max} \geq \sum_{l \in P_{(s,d)}} d_l$$

Given the requirement to establish a session, the QoS routing protocol needs to find a route with sufficient bandwidth that is at least equal to R_{bw} and an end-to-end delay that is at most equal to D_{max}. We define a path with the minimum number of hops, yet satisfying the required QoS metrics defined above as an *optimum* path.

For wireless sensor networks, we assume fixed, homogeneous, and energy-constrained sensor nodes that are randomly deployed in a fixed size sensor field. Each sensor has a transmission range (r) measured in meters. Although sensor nodes are assumed stationary, the tracked phenomena are allowed to move. We assume that contention between sensor nodes is solved by the MAC layer. Position awareness of sensor nodes is important since data collection is based on the location. Currently, it is not feasible to use Global Positioning System (GPS) hardware for this purpose. Hence, it is favorable to have GPS-free solutions for the localization problem in WSNs [121]. As the information collected by various sensors may be correlated, redundant, and/or of different qualities, we assume that sensors are divided into, possibly overlapping, groups such that each group is sensing the same phenomenon. Let Θ be the set of these groups, and let the set of members of a group $g \in \Theta$ be S_g. We denote by node 0 the BS node.

A meaningful metric to maximize in WSNs is the *network lifetime*. In the literature, network lifetime has often been defined as the time for the first node to die (run out of energy), or as the time for a certain percentage of network nodes to die such as in the case of densely deployed WSNs; thus the quality of the system is not affected until a significant amount of nodes die as adjacent nodes record identical or related data. Hence, the lifetime of the network is the time elapse until some specified portion of the nodes die. These definitions may not satisfactorily capture the intuition behind the concept of network lifetime nor capture the application-specific feature of the WSNs. This can be explained as follows. If few nodes in strategic positions die, the network could become disconnected and become unusable, hence there is no meaning using the death of percentage of nodes as a definition of network lifetime. Conversely, defining network lifetime in terms of the time for the first node to die is often pessimistic, since it is very likely that the surviving nodes remain connected. We will use a more realistic definition of the network lifetime in Chapter 5. For ease of reference, Table 2.1 summarizes notations used in the rest of the dissertation. We now introduce some useful definitions that will be used in this dissertation.

Route Lifetime: Wireless links on the route $P_{(s,d)}$ can break down due to node mobility, nodes turning ON/OFF, or environmental problems. We define the route[1] lifetime as the time until a node or a link in this route fails. To find the lifetime of a route, we must consider all the links in the route separately because a break in any of the links will break the route. We are interested in finding the lifetime of the route $P_{(s,d)}$. Let ρ_l be a random variable that represents the lifetime of a link $l \in P_{(s,d)}$ and let the random variable Υ represents the lifetime of the entire route. Hence,

$$\Upsilon = \min_{l \in P_{(s,d)}} (\rho_l)$$

Denote by η_l the probability of failure of the lth link in $P_{(s,d)}$. Assuming statistical independence

[1] Without causing confusion, the terms path and route are used interchangeably.

Table 2.1 System Model Notations used in MANETs.

Notation	Meaning
A	the network service area (m^2).
N	the number of mobile nodes in the network area A.
Z	the number of possible zones in VGA architecture.
Δ	maximum node degree in the network.
t, δt	time and smallest increment in time, respectively.
T	a window of time sufficiently large to capture cluster dynamics.
W	the number of time intervals used within T; $W = \frac{T}{\delta t}$.
$Y_i(t)$	a binary indicator, which is 1 iff node i is a CH at time t.
$Q_i(t)$	the fraction of time node i remains as a clusterhead during the time window T ending at time t, $0 \leq Q_i(t) \leq 1$.
$B_i(t)$	the remaining energy in node's i battery at time t.
$H_i(t)$	the number of times a node i served as CH in T till time t.
$C(t)$	clusterheads cardinality at time t (i.e., size) of the MCDS.
$S_i(t)$	the stability of node i being as a CH, at time t.
$S(t)$	Network stability at time t.
$v_i(t)$	average speed of node i at time t (m/s).
$D(t)$	the degree of sharing among nodes to serve as CHs in the network until time t.
$E_i(t)$	the Euclidian distance between a mobile node i and the center of the zone at time t.
$L(t)$	The clusterhead load sharing until time t.

between links, the probability of route failure, ϕ, can be found as follows

$$\phi = 1 - \prod_{l \in P_{(s,d)}} (1 - \eta_l)$$

The probability η_l can be obtained if the lth link lifetime distribution is known. Assume that route $P_{(s,d)}$ has n hops and that random variables $(\rho_1, \rho_2, ..., \rho_n)$ are independent. If the link lifetimes are *iid* process with probability distribution and density functions given by $F_l(t)$ and $f_l(t)$, respectively, then,

$$F_\Upsilon(t) = (1 - F_l(t))^n \ , \quad f_\Upsilon(t) = n f_l(t)[1 - F_l(t)]^{n-1}$$

Network Connectivity: The network ceases to be *usable* if its communication graph becomes disconnected. We say that a network graph is connected if there is at least one path between every pair of mobile nodes. A network may become disconnected when the network becomes partitioned. This could happen when the mobile nodes move in groups, for example. Let $\mathcal{U}(t)$ be the number of CH node pairs that are connected by one or more routes consisting of single or multiple links in the network at time t. Let $n(t)$ be the cardinality of the largest connected component of CHs at time t. The average network connectivity, $\mathcal{ANC}(t)$, at time t can be defined as:

$$\mathcal{ANC}(t) = \frac{\mathcal{U}(t)}{\frac{n(t)(n(t)-1)}{2}} \quad , \quad n(t) = \sum_{j=1}^{N} Y_j(t) \tag{2.1}$$

and $Y_j(t)$ is defined in Table 2.1.

Network Stability: An ad hoc network is considered stable if the frequent topological changes can be made transparent to the users. That is, node movement cannot result in a severe degradation of the network performance in terms of route lifetime and service quality. We focus on the stability of VGA. We say that VGA is stable if the topology updates of VGA are rare. This happens when VGA members (i.e., CHs) remain in that state for relatively long time. CH role in a zone should be changed only when necessary. Typically, a CH role is changed either due to CH node mobility or due to drastic changes in the status of CH. If a certain CH is performing well, replacing it will result in unnecessary changes in VGA, which may incur extra overhead[2] and result in a less stable system. In VGA, as long as each zone maintains at least one node, the virtual architecture will not change. Frequent CH changes decrease a zone stability. A higher zone stability indicates that nodes change the role of CH at reasonably low frequency. Network stability is affected by CH nodes stabilities as shown later in Chapter 3.

2.3 VGA Clustering Approach

In this section, we present the Virtual Grid Architecture (VGA) clustering approach. The main objective of VGA clustering approach is to create a simple and stable rectilinear virtual topology, namely VGA, on which the routing and network management functions can be performed easily and efficiently. The first step in creating VGA is the network zoning or clustering, i.e., creating the zones or the clusters, which is discussed next.

2.3.1 The Zoning Process

Our zoning process starts by dividing the network area into disjoint, adjacent, fixed size, and regular (symmetric) shape zones. The ideal zone shape would be a circle, which makes it possible to define a circular (radial) zone around a mobile node. However, this will leave a number of network parts outside the coverage, so the optimal shape of the zone is hexagonal. To create a simple rectilinear virtual topology, however, we select the zones to be square in shape[3]. Since zones are fixed, sophisticated algorithms for zone management are not needed. Formally, the network area, defined by the coordinates $(0.0, 0.0) - (MAX_x, MAX_y)$ is divided into fixed-size and equal square zones. Let Z be the set of those

[2] The overhead includes electing a new CH, migrating the routing table to the new CH, and taking care of queued packets at the old CH.

[3] Extension of the virtual topology to other shapes is possible with the following tradeoffs. If the zone shape is a hexagon instead of square, the routing function will be more complex. If, on the other hand, the zone shape is linear or triangular, the resulting virtual topology will be very simple but less efficient.

zones. Let $z_i \in Z$ be the i^{th} zone, and denote its initial and final coordinates by $(x_i, y_i) - (x_f, y_f)$. Each zone z_i has a unique address (ZoneID) which can be the ordered pair consisting of the zone row and column numbers, respectively. Each mobile node is a member of one of those zones and its zone membership is determined based on its location in the network area.

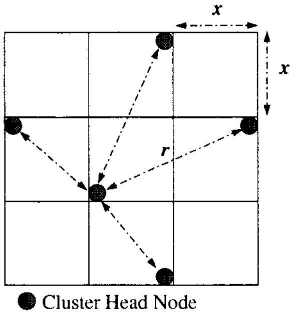


Figure 2.4 Selection of zone side length.

In homogeneous MANETs, where all nodes have the same transmission range, r, the zone side length x is chosen such that two mobile nodes in adjacent horizontal/vertical zones and located *anywhere* in their zones, can communicate with each other directly. A necessary and sufficient condition for this communication to happen is to set the zone side length (x) to $\frac{r}{\sqrt{5}}$ (see Figure 2.4). Note that diagonal communication between zones which are corner adjacent is possible, but in this case the zone size will be smaller since $x = \frac{r}{2\sqrt{2}}$, hence increasing average route length and routing cost as shown in Chapter 3. An example of the fixed zoning process applied to homogeneous networks as well as the resulting virtual topology is shown in Figure 2.5(a). Given the current coordinates of a node as (a,b) we can calculate its zone as:

$$zone = \lfloor(\frac{a}{x})\rfloor + Z_y * \lfloor(\frac{b}{x})\rfloor$$

where Z_y is the number of zones on the Y axis as will be explained in the next chapter.

In heterogeneous MANETs, mobile nodes have different transmission ranges. For the sake of simplicity, but without loss of generality, we assume that mobile nodes are of two different transmission ranges: r_s for Short Range (SR) nodes and r_l for Long Range (LR) nodes. The transmission range is a function of the energy level at the mobile node in each case. The extension to more than two transmission ranges is possible and follows the same procedure outlined in this section. The transmission ranges play important role in the zoning process and in creating the virtual topology. We need to provide intra-zone and inter-zone communications while using minimum transmission power. Intra-zone communication refers to the communication between nodes inside a certain zone while inter-zone communications refers to communication between nodes in different and neighboring zones. A zone $z \in Z$ may only have nodes of short, long, or both long and short transmission ranges. Initially, the network area is divided into large zones where the zone side length x_l is chosen such that two long range mobile nodes in adjacent horizontal/vertical zones and located anywhere in the zone can communicate with each other directly. Therefore, x_l is selected as $x_l = \frac{r_l}{\sqrt{5}}$. If the zone has only short range nodes, the zone is divided into four[4] *subzones* where the short range, r_s, is chosen such that each subzone side length, (x_s), is set as $x_s = \frac{x_l}{2}$. The number of layers in the virtual topology is determined by the number of transmission ranges in the network. Figure 2.5(b) shows an example of the zoning process in heterogeneous networks and the resulting virtual topology. Note that in Figure 2.5(b), the left side zones have both SR and LR nodes while the right side zones have only SR nodes. Therefore, each of the right side zones is divided into four subzones to allow direct communication between any two short range mobile nodes in adjacent horizontal/vertical subzones. One salient feature of our zoning process is that the resulting virtual grid limits the control overhead to communication between CHs and in the associated vertical and horizontal directions only.

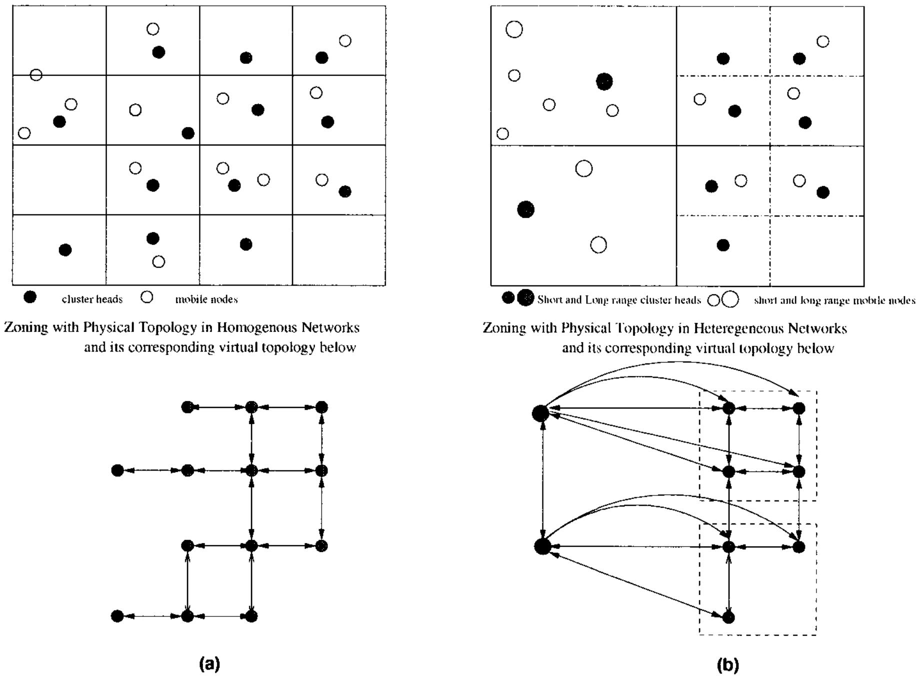

● cluster heads ○ mobile nodes
Zoning with Physical Topology in Homogenous Networks and its corresponding virtual topology below
●● Short and Long range cluster heads ○○ short and long range mobile nodes
Zoning with Physical Topology in Heteregeneous Networks and its corresponding virtual topology below
(a)
(b)

Figure 2.5 The Fixed zoning process in both (a) homogeneous networks (i.e., identical transmission range) and (b) heterogeneous networks (i.e., variable transmission range): An example.

Since both node types are allowed to move freely, it may happen that an LR node leaves its zone to a neighboring zone which only has SR nodes. If the LR node is the only node of its type in the old zone and the rest is SR nodes, then a split operation takes place in the old zone. The LR sends

[4] In general, if we have m transmission levels, then larger zones will be divided into subzones using binary form, i.e., $x_s^{(k)} = \frac{x_l}{2^k}, k = 1, 2, 3, \ldots, m$ going from large to small zones.

out a beacon signal[5] to indicate a split process in its old zone such that the zone is divided into four subzones and a SR CH node is selected inside each subzone. Furthermore, if the new zone of that LR node has only SR nodes in it, i.e., it is originally divided into four subzones, a merge process will occur. This happens when SR nodes in these subzones receive a beacon signal from the LR node that signals the joining of an LR node to these subzones. The beacon signal sent by the LR node to the SR nodes in the new zone serves as an indication for a merge process. Upon hearing the beacon signal, all SR nodes in the new zone will set this LR node as their new CH and abandon their current SR CH nodes. In general, the split and the merge processes take place independent of each other, and depending on the situation imposed by node movements. Figure 2.6 gives an example of a split process and a merge process that may happen in heterogeneous MANETs. The situation in this figure resembles the case of LR node movement (node A in the figure) we just discussed, where a split process takes place in the old zone and a merge process in the new zone. It is important to note that in homogeneous MANETs, the split and the merge processes are not necessary as all nodes exhibit the same transmission range, and hence all zones are fixed and retain the same size regardless of node movement.

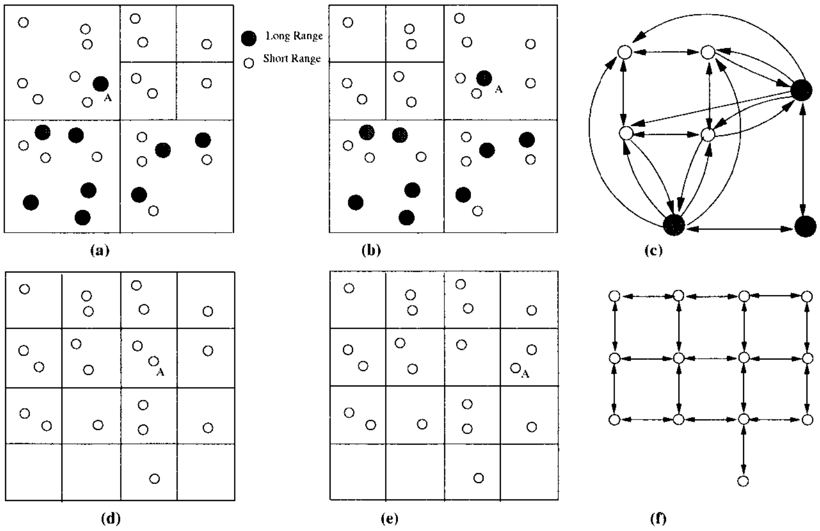


Figure 2.6 Example of the zoning process applied to the network area and the resulting virtual topology in both homogeneous and heterogeneous MANETs (a) zoning in heterogeneous network (b) split and merge processes due to node movement (c) the new virtual topology (d) zoning in homogeneous network (e) topology is fixed with node movement (f) no change in virtual topology.

In VGA, channel management for intra-zone and inter-zone communication is performed as follows. Mobile nodes inside a certain zone, which are not acting as CHs, communicate directly using the IEEE 802.11 MAC protocol mechanism. However, channel communication management between different

[5] We assume that all messages exchanged between mobile nodes include node types, i.e., SR or LR node.

zones (CHs) is controlled by using code division multiple access (CDMA). The assignments of codes to zones is according to [49], where neighboring zones use different spreading codes in order to reduce interference and to enhance spatial reuse of channels. A modified MAC protocol can also be associated with the VGA protocol as follows. The modified MAC protocol is a combination of carrier-sense multiple access (CSMA), time-division multiple access (TDMA), and direct-sequence spread spectrum (DSSS). The application determines which MAC protocol is used to send each message based on the constraints of the routing protocol. For example, if the packet is an advertisement or a join-request message, it is transmitted using a CSMA approach and using a dedicated frequency channel for this purpose. If it is a data message being sent to the clusterhead, it is sent using a TDMA slot with the DSSS code specified by the CH. Using such an approach, the MAC protocol is always chosen such that it reduces energy dissipation and minimizes collisions (e.g., using DSSS on top of TDMA or using CSMA to reduce the number of collisions).

2.3.2 CH Election

After the zoning process is finished, a CH election algorithm is executed in each zone. Network access within each zone is attained through the elected CH. The motivation is that since there is no fixed infrastructure with a high-energy or high performance node that can act as a cluster-head, one of the mobile nodes must take on this role in each cluster such that this role is also shared among nodes. The CH is elected via a completely distributed election process. Also, the CH role inside each zone can be changed dynamically or periodically. Varying the election period or fixing it are two viable options. Fixing this period results in simpler operation but may not be optimal. Varying the period in an adaptive manner can result in better behavior but may require more computation, overhead, and power consumption. The tradeoff between these two options is studied and the results are discussed in Section 3.4. The optimal length of the CH period depends on a number of parameters such as cluster load capacity, energy level, speed, and number of mobile nodes in the zone. Note that CH election periodicity helps balance the nodes' load distribution, achieves fairness, and provides fault tolerance against single node failure. A CH may also change its role dynamically, e.g., if it moves out of the zone or if it fails for any reason. It can also ask to be replaced on demand if its battery is about to be drained or if it is overloaded. This dynamic CH role change can result in a more adaptive clustering scheme. In the following, we discuss the two scenarios that can be used for CH election, namely periodic and adaptive CH election, in more details. In both scenarios, the CH election algorithm is fully distributed and is executed in each zone independently, i.e., the CH election protocol is concurrent in the sense that it runs all over the network at the same time and many zones can elect their CHs simultaneously.

Periodic CH election: In case of periodic CH role change, we define an election round in each zone as the time period in which all nodes, that are eligible, have served as CHs. Each election round is, in turn, divided into one or more election periods. Each mobile node decides whether it will become a CH for the current election period based on an eligibility factor (EF). The eligibility factor of a node i, EF_i, to serve as a CH at time t is calculated as:

$$EF_i(t) = a_1 e^{-v_i(t)} + a_2(1 - Q_i(t)) + a_3 B_i(t) + a_4(1 - S_i(t)) \tag{2.2}$$

where $v_i(t)$ is the speed of node i at time t, $B_i(t)$ is the remaining energy at ith node battery at time t, $Q_i(t)$ is the fraction of time node i served as CH in a large window of time (see Section 2.4.4.1) that reflects the energy level difference between the previous period and the current period of that node, $S_i(t)$ is the Euclidean distance of node i with respect to the center of a zone at time t, and a_1, a_2, a_3, a_4 are scaling factors that reflect the importance of each parameter such that $\sum_{i=1}^{4} a_i = 1$ and $0 \leq a_i \leq 1, i = 1, \ldots, 5$. The node that has the highest value of EF will elect itself as the CH for the current period and will send a broadcast message for nodes in its zone. That is, if a zone z has a nonempty set of nodes m, the CH is selected as follows:

$$CH = arg \max_{i \in m}(EF_i)$$

The rationale behind the eligibility factor is that the most eligible nodes should be selected to act as CHs for the longest time possible. A simple algorithm that performs CH election in a non-empty zone $z \in Z$, which currently has a nonempty set of nodes m is shown in Figure 2.7. In the algorithm, each node in zone z calculates its EF and broadcasts it to other nodes in the zone. A simple comparison at each node determines which node has the maximum EF, i.e., the node that becomes a CH. Then, the node with the maximum EF will broadcast its ID declaring itself as the CH. In heterogeneous settings, the broadcast message also includes a field for the node type (SR or LR) that is used for the split and merge processes as was indicated earlier. Note that a mobile node inside a certain zone can know the location (membership) of other nodes in the zone through different methods. For example, nodes can advertise their own coordinates, i.e. zone ID, through beacon signals. After successful election, each mobile node that has elected itself as a CH for the current period will broadcast its ID as an advertisement message to the set of the mobile nodes in its zone. Following each new election, a *handoff* procedure is executed after which the new CH inherits the routing cache of the retired CH. However, in order to minimize delays incurred due to CH role exchange, we allow the retired CH to transmit any remaining queued packets if it is not out of service. Otherwise, new CH is responsible for routing these packets. A CH will act as such for a certain period of time. After the end of this period, another node in the zone will be elected as the CH.

Dynamic CH election: Under this scenario, a CH in each zone is initially selected based on equation (2.2). Unlike periodic CH election, the eligibility factor is calculated again whenever there is a need for a new CH in the zone. Whenever the current CH in a certain zone is about to leave its zone or its battery energy falls below a certain threshold, it will initiate a new CH election in that zone.

```
Data     : zone $z \in Z$
Result   : A CH for current period in zone $z$
initialization: CH-elected=false;
if (Zone $z$ not empty) then
    while (!CH-elected) do
        for (each node $j \in m$) do
            Calculate $EF_j$;
            Get $EF_k$;   $k \neq j$, $k \in m$;
            if ($EF_k > EF_j$) then
                Do not broadcast $EF_j$
            else
                Broadcast $EF_j$;
            end
        end
        // find the maximum $EF$ at each node $j$;
        for (each node $j \in m$) do
            $EF_m$ = maximum(All Received $EF's$);
            if ($EF_j = EF_m$) then
                CH = $j$;
                Broadcast Node-ID($j$);
                CH-elected=true;
            end
        end
    end
end
```

Figure 2.7 Cluster head Election Algorithm

The most eligible node (based on the calculation of a new EF) will then be elected as the new CH in that zone. When the energy level of all nodes in a certain zone falls below the selected threshold, the election algorithm demands that nodes in that zone will rotate the CH role among themselves based on their IDs in ascending order. Each node will act as a CH until the node is out of service due to battery energy depletion.

On-demand CH election: In this strategy, a CH node may ask to be replaced on demand if its battery is about to be drained or if it is overloaded or the node needs to leave the network or switch off. This strategy can be mixed with the previous two strategies for exceptional events where these events are rarely used.

Although the concept of CH election is not a new concept, our zoning approach simplifies its operation. Furthermore, our CH election algorithm results in a more stable topology. Note that if a certain zone has no nodes, it will not have a CH, which may result in broken links in the virtual topology. It could also happen that the zone may only have a single node which by default acts as a CH.

2.3.3 Stimulating Node Cooperation in MANETs

In the CH election process, we assumed a cooperative environment, in which all mobile nodes participate towards the CH election process. When the cooperative assumption is relaxed, such as in the case of selfish users, a conflict emerges between local energy conservation and network communications performance. From this perspective, the assumption of a cooperative network environment is not always justified. As such, a local, rational mobile node may prefer to avoid network participation, i.e. act as a CH, in order to better satisfy its energy performance goals. If all mobile nodes in a zone employ this reasoning, then the CH election protocol might fail; hence the multihop delivery capability of the network will be severely limited. Moreover, if no user cooperates in the CH election in a certain zone, it will result in a loss in connectivity of this zone to the rest of the network.

Most of previous work on MANETs has implicitly assumed that nodes are cooperative, i.e., whenever a node receives a request to relay traffic or become a CH, it will always do so. However, this approach ignores the user specific requirements or attitude. Consider, for example, an airport lobby, a classroom, or a conference hall with a gateway node to the Internet. A number of users might form a MANET to access the gateway node. The user physically near the gateway node will end up relaying most of the traffic. However, since this user views his energy resource as being limited by battery life, he may not feel inclined to act as a CH (relay traffic) for other users. As such, user's behavior will impact the system performance driven by his application needs or physical constraints. On the other hand, if we assume that each user wishes to maximize his throughput, it may be in his best interest to be cooperative and accept the CH role or relay traffic for another user as other users will then honor his cooperation at a later time, i.e., his connection request will not be blocked. The question is how can we stimulate the user or enforce him to cooperate.

The problem of non-cooperative mobile nodes in MANETs has been addressed in very few works [130, 131, 132, 133, 134]. In [131], non-cooperative nodes are viewed as malicious, and methods to identify misbehaving users and to avoid routing through these nodes are presented. In [130, 132, 133], simple rules are used to determine on a packet-by-packet basis whether a user should forward other nodes traffic or not. In particular, in [130, 132], the authors introduce a virtual currency called nuglets. Every network node has an initial stock of nuglets. Either the source or the destination of each traffic connection use nuglets to pay the relay nodes for forwarding data traffic. Packets sent by or destined to nodes that do not have a sufficient amount of nuglets are discarded. Notice that by allowing source or destination to charge can under- or over-estimate the packet price. Moreover, both proposals require a tamper-proof hardware at each node so that the correct amount of credit is added or deducted from the node. As a result of this requirement, both proposals may not find wide-spread acceptance. In [133], source nodes pay as many battery units as the estimated number of nodes on the path to the destination, and makes relay nodes earn as many battery units as the number of forwarded packets. In [134], a game-theoretic approach for routing in MANETs, called Ad hoc Vickrey, Clarke, and Groves (Ad hoc-VCG) that consists of greedy and selfish agents was considered. Those agents accept payments for forwarding data for other agents if the payments cover their individual costs incurred by forwarding data. Ad hoc-VCG is a reactive routing protocol, which may guarantee that routing is done along the most cost-efficient path by paying to the intermediate nodes a premium over their actual costs for forwarding data packets.

Most of these approaches share the following critical concerns, that still need to be solved: (i) The packet-by-packet paying system imply a significant communication overhead and implementation complexity, (ii) They all assume the users are of the same behavior or of the same class, (iii) They did not pay attention to the fairness issue in routing when some nodes do not get any reward due to some reasons, e.g., location, (iv) none of these approaches dealt with cluster-based networks where the role of clusterhead is critical for the operation of the network as a whole, and (v) some proposals require tamper-proof hardware at nodes, while others assume mobile nodes have secure access to a trusted third party through Internet, for example.

In this subsection, we develop a simple solution to a non-cooperative MANET environment that addresses the above concerns. Our reputation-based approach stimulates and enforces nodes to cooperate at two levels: intra-zone (CH election process) and inter-zone (relaying traffic to other zones). In both cases, the decision of a mobile node to cooperate depends on many parameters, e.g., node's location, node's energy constraints, node's mobility pattern, and its particular needs. Therefore, we associate with each node a parameter, called ***willingness***, that refers to the node's willingness to cooperate. The willingness of node i to act as CH at period p, denoted by, w_i^p, takes the values [0,1], where a value of 0 refers to a completely selfish node while a value of 1 refers to a completely cooperative node. Note that the value of w_i^p can be a constant or it can be dynamically set by each node at the beginning of each election period or at the beginning of a new routing session. The value of w_i^p may depend on the status of that node at the time of decision, e.g., its current energy budget and its attitude.

Our objective is to allow nodes to be selfish if they need to be so, and study the impact of their behavior on the system performance reflected by the CH election process and the traffic relaying process. In the CH election process, we want to stimulate the most eligible node to always accept the role of CH if it wishes to maximize its throughput. Let Z be the set of resulting zones. Let $N(z)$ be the number of nodes in zone z, $z \in Z$. Let E_i^p be the power spent per byte by node i in transmitting packets for other nodes in its zone during period p. We assume that each node has a memory of whether it has been helped earlier by each node in the zone. Hence, a node will always remember the favor done to it by other nodes in the zone, as well as the selfishness of other nodes. To do that, we associate with each node two more parameters: ***take*** and ***give***. The parameter ***take*** intuitively reflects the amount of help that a node has received from other nodes in the zone relaying its messages. On the other hand, the parameter ***give*** reflects the amount of help that the node has rendered in relaying messages for other nodes in the zone while being a CH. The proposed algorithm attempts to balance the amount of takes and gives at each node in the network. Given a value of w_i^p, a node i is more willing to act as CH if it has received more help than it has given. Conversely, if a node has been generous in the past without receiving a commensurate amount of assistance from other nodes, then it is inclined to reject the CH role. As the value of w_i^p decreases, nodes tend to behave more selfishly.

2.3.3.1 Intra-Zone Cooperation: CH election

The objective of intra-zone cooperation process is to ensure that the most eligible node will always accept the CH role. When a node serves as a CH, it will maintain a table with entries for all nodes that have received help (node IDs), the amount of the received help (take), and the amount of the rendered help in previous periods (give). At the end of the current election period, the CH will broadcast an INFO message that contains the information in this table to all nodes in the zone. The values in this table will be used to calculate the eligibility factor for the next period as follows. At the beginning of the election process, each node in the zone will receive the EF values of all other nodes. Before doing the comparison, the value of the EF received from the generic ith node will be modified in accordance with the difference between the amount of help the node received and the amount of help the node rendered in the network, respectively, during the previous period(s). Let $take(i,p)$ and $give(i,p)$ be the amount of take/give parameters of the generic i-th node at the beginning of period p. Then, the new eligibility factor will be calculated as,

$$EF_j = EF_j * (take(j,p) - give(j,p)) \quad j \neq i, j \in N(z) \tag{2.3}$$

The values of take and give parameters are received in each INFO message. The second term will increase the value of EF for those nodes who received more help than what they give, and hence need to help in turn. In order to ensure that these nodes who have been acting selfishly will not receive more help, a CH node will forward traffic for non-CH node i in its zone if the following condition is met:

$$give(i,p) > (1 - w_i^p).\{take(i,p) + Msg\_size * E_i^p\} \tag{2.4}$$

where Msg_size is the size of message to be transmitted in bytes. The above condition forces nodes to always try to provide help (i.e., have credit) in order for its packets to be forwarded in turn. This credit will be obtained if the node has served as a CH in the previous periods.

2.3.3.2 Inter-Zone Cooperation: relay traffic

The objective of inter-zone cooperation process is to ensure that intermediate CH nodes will always help in relaying traffic for other CH nodes in the network. Assume that a session is generated between a source-destination CH pair (s-d) and that the set of available routes between this pair is $\mathcal{R}$. Consider the minimum energy cost path $r \in \mathcal{R}$. Let $e(k,r)$ be the amount of energy per byte spent by node k on the route r in relaying traffic to the next node on r. Let $t(k,r)$ and $g(k,r)$ be the amount of take/give parameters of the generic k-th node in route r. Let $N(r)$ be the set of CH nodes on the route

r, respectively. When a source node s initiates a request and the request is accepted by the relay nodes on r, the node can forward its traffic for the whole session. For each session accepted for any source node s on the route r, the ith relay node $i \in N(r)$ will store the amount of help it gives to node s as follows:

$$give(i,r) = give(i,r) + \{Msg\_size * e(i,r)\} \quad \forall\, i \in N(r) \tag{2.5}$$

At the end of a session, each relay node will attach to the last ACK message, that is sent back to the source node, the value of the amount of help the node rendered in forwarding the traffic of this session. Accordingly, the source CH node will update its take parameter as follows:

$$take(s,r) = take(s,r) + \sum_{i \in N(r)} give(i,r) \tag{2.6}$$

A relay CH node i will accept forwarding traffic for a source CH node s if the node received help from the network more or equal to what it gives, i.e., the following condition should be met:

$$take(i,r) - give(i,r) >= (1 - w_i^r).\{Msg\_size * e(i,r)\} \tag{2.7}$$

Hence, each node will be trying to repay the network whatever it takes from the network (debt). As a result, nodes are stimulated or enforced to cooperate for their own best interest.

2.3.4 A Simple Power Control Scheme

In this section, we present a simple power control scheme that is augmented with VGA. A power control scheme is particularly needed in heterogeneous MANETs to ensure network connectivity while minimizing power consumption and interference in the network. The scheme determines the power levels required for inter-zone and intra-zone communications. Since we have two transmission ranges, four types of communication between mobile nodes are defined (Short→Short(S-S), Long→Long(L-L), Long→Short(L-S), and Short→Long(S-L)). Note that a LR node can communicate directly with all other nodes. However, a communication problem arises when a SR node wants to communicate with a LR node inside a zone or in a neighboring zone. To solve this problem, we devise a simple power control mechanism that works as follows. Assume the distance between two mobile nodes is d. The transmitted signal strength is known to decrease with the distance d according to the relation [53],

$$P_r = G\frac{P_t}{d^\alpha}$$

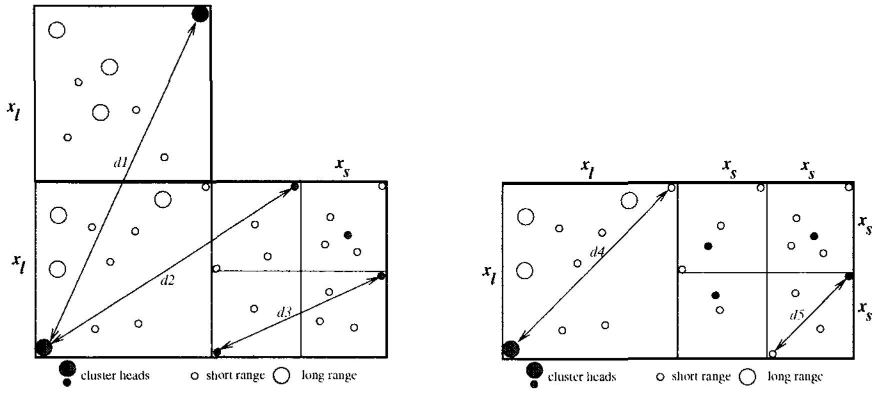
where P_t, P_r are the transmitted and received signal power, respectively, α characterizes the steepness of the decrease in the signal power and it assumes values in the range $2 \leq \alpha \leq 4$ in a homogeneous medium, and G is a constant. For convenience, we define the ratio of the transmitted signal power of any mobile node to that of LR node in a homogeneous medium, denoted by RTP, as

$$RTP = \frac{P_t}{P_{max}} = (\frac{r}{r_l})^\alpha$$

where P_t and P_{max} are the transmitted signal power of the target mobile node and LR node, respectively and $r = r_s$ or r_l for SR nodes and LR nodes, respectively. Figure 2.8 shows the different cases of intra-zone and inter-zone communication. The distances d_1, d_2, d_3 represent the maximum distance separation between CHs for inter-zone communication, while d_4, d_5 correspond to the maximum distance separation between mobile nodes and CHs for intra-zone communications. From Figure 2.8, we find the worst case values of RTP for different types of communication. The results are summarized in Table 2.2. The values between parentheses are the worst case values of RTP corresponding to each type of communication. They were obtained through simple derivations. For example, if the sender is a LR node, then it uses its maximum power (P_{max}) to communicate with another LR range in a neighboring zone. An LR node can use a reduced power P_{red} to communicate with an LR mobile node inside a zone, which corresponds to a distance of $\sqrt{\frac{2}{5}}r_l$. Assuming that the receiver sensitivity for all mobile nodes is the same, then

$$P_{red} = \frac{4P_{max}}{25} < \frac{P_{max}}{6}$$

This means that transmission power can be reduced by a factor of 6. Hence, the total power needed for communication can be reduced by this factor which results in prolonging the lifetime of the batteries of mobile nodes.



(a) Inter–Zone Communication

(a) Intra–Zone Communication

Figure 2.8 (a) Inter-zone communication (b) Intra-zone communication in VGA.

Table 2.2 Summary of Intra-zone and Inter-zone communication ranges and RTP values

Type	S-S	S-L	L-S	L-L
Intra-zone range(RTP)	$\frac{r_l}{\sqrt{10}}(0.1)$	$0.63r_l(0.39)$	$\sqrt{\frac{2}{5}}r_l(0.39)$	$\sqrt{\frac{2}{5}}r_l(0.17)$
Inter-zone range(RTP)	$\frac{r_l}{2}(0.25)$	$0.81r_l(0.65)$	$\frac{3}{2\sqrt{5}}r_l(0.4)$	$r_l(1.0)$

Determining the optimal transmission range to reduce the power levels has been an issue of some research work lately. In a recent work [55], the authors used a concept called *contention index* to improve the power performance of MANETs. The basic idea of [55] is that each mobile node maintains an optimal table of the values of the contention index. Each node uses this optimal value of the contention index to set its transmission range dynamically for the best performance of the network in terms of the network capacity and power efficiency. This approach can be applied to further extend the power efficiency of our scheme, but at higher computational cost.

2.4 Routing over VGA

In recent years, numerous routing protocols were developed for MANETs. In particular, the Ad hoc On-Demand Distance Vector routing protocol(AODV) [57], the Dynamic Source Routing protocol(DSR) [58], and the Temporary-Ordered Routing Algorithm(TORA) [59] all demonstrated the ability to route data packets efficiently. A recent review of traditional routing protocols for MANETs can be found in [60, 61]. In this section, we investigate two methods to discover paths between a source-destination pair over VGA. The first method is based on the well-known *on-demand* routing techniques but with an original twist: packet broadcasting is constrained to only at most four neighbors. The second method is based on the concept of *transitive closure* from discrete mathematics. Using this concept, a CH node can test for the existence of a path and find that path by performing local computations. Routing over VGA is simple where packets are routed through the set of CHs and in the associated vertical and horizontal directions only. We have not found in the literature any routing technique that exploit the concept of transitive closure to perform routing in MANETs.

The first phase of the routing problem is to find a path between a source-destination pair (s,d), which is a sequence of edges that link s to d on the virtual rectilinear graph G. The second phase is to route packets over such a path. Several obvious advantages of a grid-based routing can be identified. First, a well-defined coordinate system for the network area will provide corner reference for easy routing. Second, routing from one node to another can be performed using multiple routes. It is well known that multiple routes provide robustness in the face of frequent topological changes which occur frequently in MANETs. Nodes within the same zone can communicate directly. If the destination node is outside the zone, the source node contacts the zone CH to find a path and then forwards packets to the destination node over that path.

It is worth mentioning that VGA is similar in structure to the grid network in wireline networks (e.g., Manhattan Street Network (MSN)). Several routing schemes have been proposed for this routing structure in the wireline networks, e.g., Store-and-Forward, Deflection routing, Hot Potato routing, and vertical routing. These routing strategies, governed by simple routing rules that exploit the regular structure of the network, rely on the fact that decisions must be taken at every node for a packet to go from its source to its destination. The protocols mentioned above are not suitable for applications in MANETs because of their inherent advantages and disadvantages that may not suite the nature of ad hoc networks. For example, these protocols assume that all links are always available, while in VGA links can be added/deleted over the lifetime of the network. However, we believe that these algorithms can be implemented on top of VGA with necessary modifications.

2.4.1 On-Demand Routing Approach

A simple on-demand packet forwarding scheme (OD-PFS) can be easily implemented over VGA. The OD-PFS scheme can be implemented as follows. The standard four directions (North(N), South(S), West(W), East(E)) are used for simple packet forwarding in the resulting virtual grid, VGA. Those directions can be encoded using a 2-bit representation in the packet header such that (00-11-10-01) correspond to (N-S-E-W) directions, respectively. Note that opposite directions have complementary codes, which simplifies route traversal over VGA. OD-PFS builds routes using a route request (RREQ)/route reply(RREP) query cycle. The RREQ packet collects the IDs of the intermediate CH nodes and directions, and stores them in the path field in the packet header. Each CH node maintains a list of its current zone members. The detailed operation of OD-PFS is illustrated as follows:

1. When a source node wishes to communicate with a destination node, it first needs to know the destination location (e.g., Zone ID or CH ID). Hence, the source node forwards its request to its CH. In turn, the source CH node issues a location request (LREQ) messages to its neighboring CHs[6]. If any CH knows the destination zone ID, it sends back a response using location reply (LREP) packet which contains the destination zone ID. Otherwise, it will send the LREQ to all neighbor CHs nodes. In the worst case, the LREQ will reach the destination zone, which then returns its zone ID. However, if intermediate nodes happen to know the location of the destination mobile node, the operation will yield a much faster reply.

2. If a node learns that the destination node is in the same zone as itself[7], it initiates communication directly with the node. If traffic is destined outside the zone, the source CH node initiates the route discovery process at the network layer if it does not already have a route to that destination. The source CH broadcasts a route request (RREQ) packet to its possible four neighboring CH nodes.

3. Each intermediate CH mobile node receiving the RREQ will check for duplicate RREQ. The RREQ contains the most recent sequence number for the destination of which the source node is aware. A CH node receiving the RREQ may send a route reply (RREP) if it is either the

[6] Before that, the CH will also check locally with its zone members through a Local Location Request (LLREQ) about destination zone ID

[7] The source node knows the IDs of other nodes in its zone which are refreshed at every CH election phase due to ID exchanges.

destination or if it has a route to the destination with corresponding sequence number greater than or equal to that contained in the RREQ. If this is the case, it unicasts a RREP back to the source CH by backward traversal of the route, and using the reverse directions. Otherwise, it appends its address and re-broadcasts the request in the three remaining directions towards the three neighboring CHs (excluding the receiving direction). Nodes keep track of the RREQ's source IP address and broadcast ID. If they receive a RREQ which they have already processed, they discard the RREQ and do not forward it.

4. As the RREP propagates back to the source CH, nodes set up forward pointers to the destination. Once the source CH node receives the RREP, it may begin to forward data packets to the destination. If the source later receives a RREP containing a greater sequence number or contains the same sequence number with a smaller hopcount, it may update its routing information for that destination and begin using the better route. The above operation may result in multiple disjoint paths being discovered.

5. As long as the route remains active, it will continue to be maintained. A route is considered active as long as there are data packets periodically travelling from the source to the destination along that path. Once the source stops sending data packets, the links will time out and will eventually be deleted from the intermediate node routing tables. If a link breaks while the route is active, the node upstream of the break propagates a route error (RERR) message to the source node to inform it of the now unreachable destination(s). After receiving the RERR, if the source node still desires the route, it can reinitiate route discovery.

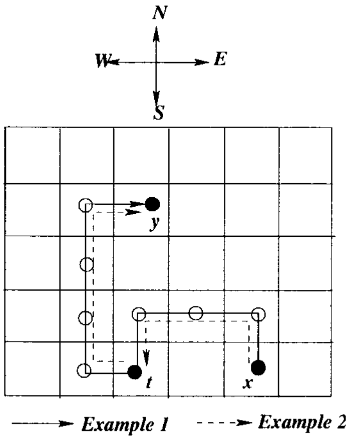


Figure 2.9 On-Demand Packet Forwarding/Routing over VGA: An example.

To illustrate the operation of OD-PFS, consider the following two examples covering the cases when the intermediate nodes do not know a path to the destination and when some of them do, respectively.

For the first case, consider a certain source node (x) requesting communication to a destination node (y) as shown in Figure 2.9. Node x initiates a route discovery by sending the RREQ packet. Assume that the complete route traversed by the RREQ when it learns a path to y (i.e, RREQ reaches y) is (N-W-W-S-W-N-N-N-E) from left to right. Therefore, the RREP will follow the path (W-S-S-S-E-N-E-E-S) to x which is simply obtained by reversing the direction in the path and taking the 1's complement of the direction of each hop. For the second case, assume the RREQ issued by x reaches an intermediate node, t, after four hops and RREQ contains the partial path, P_1 = (N-W-W-S). Also assume that node t has a path to y given as P_2 = (W-N-N-N-E). Then t will concatenate both paths P_1 and P_2 as one path $P = P_1||P_2$, where $||$ is the concatenation parameter. Hence, the resultant path to the destinations is P= (N-W-W-S-W-N-N-N-E) and node t will use RREP to send P back to the source node using the following path (N-E-E-S). In the case of heterogeneous MANETs, an additional 2-bit field per hop is also used to select a specific subzone in a neighboring zone that has only SR nodes and the direction of communication is from LR node to a specific SR node in that neighboring zone.

Our version of on-demand routing which acts on the virtual topology, VGA, reflects higher accuracy and faster response to network changes, and with low control overhead. This is because flooding is only performed among the set of CHs. Therefore, even when mobility is high and traffic is dense, our version of the on-demand routing protocol is still efficient. As mentioned, the process of path discovery may return many paths. This signals the use of multiple path routing or performing fault tolerant routing where one path can be used as primary, and the set of remaining paths can be saved as backup paths. However, backup paths may remain valid if the network topology slowly changes with time.

2.4.2 Transitive Closure Routing Approach

In this subsection, the concept of *transitive closure* from discrete mathematics [54] will be used to discover paths over VGA. In this approach, each CH must know about the status of all links in the network or at least partial knowledge about the link status, where we assume that each CH maintains an up-to-date image of the adjacency matrix ($A = \{a_{ij}\}$, $i, j = 1, ..., n$). The adjacency matrix reflects the link status between the set of neighboring CHs such that a value of 1 in the entry (i,j) indicates that there is a link going from CH i to CH j; and 0 if no link exists. The set of CHs exchange the adjacency matrix whenever it is only updated due to link addition/deletion[8]. In another implementation option, the CH in each zone may periodically send a broadcast message to all CHs indicating its presence. All CHs maintain a *soft-state* of other CHs, which expires if they are not refreshed within a certain time window.

Now, we illustrate how the concept of transitive closure can be used to find a path in the graph G. Let R be the relation on the set V with n elements, where n is the number of CHs in G (the size

[8] This is the implementation option used in the simulations.

of the virtual graph G). The relation R defines the one way connectivity between CHs and it can be mathematically defined as as follows,

$$R = \{(x, y)|x, y \in V,\ \ y \text{ is reachable from } \ x \ \text{ in one hop}\}$$

where (x,y) is the ordered pairs of CHs in V. The relation R is typically asymmetric on the set V, i.e., if (x,y) $\in R$, then it is not necessary that (y, x) $\in R$. Moreover, if there is a path in R from x to y, then there is such a path with length not exceeding n. The transitive closure of R, called R^*, is the union of R, R^2, ..., and R^n, i.e.,

$$R^* = R \cup R^2 \cup \ldots \cup R^n$$

Let T_R be the boolean (zero-one) matrix corresponding to the relation R, then the boolean matrix of the transitive closure is $T_R^* = T_R \vee T_R^2 \vee T_R^3 \vee \ldots T_R^n$. To compute the matrix T_R^*, successive boolean powers of T_R (or A), up to the n^{th} power are computed. Since all entries in T_R are either 0 or 1, and since each row or column in T_R contains at most four nonzero element, then each matrix multiplication operation can be done using at most $4n^2$ addition operations. This process can even be performed in n^2 bit vector operation; hence, these products can be computed using n^3 bit operations[9]. The search for the existence of a path between two mobile nodes (s,d) can be determined as follows. Suppose we compute T_R^i. If T_R^i of (s,d) is equal to 1, then a route of length i exists between (s,d). If T_R^i is not 1, then we calculate T_R^{i+1} and check for a route again. In the worst case, the transitive closure is used to determine if a path of length n between (s,d) exists or not. This is done by inspecting $T_R, T_R^2, T_R^3, \ldots, T_R^n$.

For a square network area, it can be shown that the maximum path length is $(n + \sqrt{n})/2$, which can be easily found by inspecting the path between any two nodes in opposite corners. As an example, if we assume that the network area is 600m×600m and the node transmission range is 100m, this results in a zone side length of approximately $45m$. The number of horizontal zones is 14 and the number of vertical zones is also 14. Therefore, the virtual rectilinear topology will have at most 14×14 vertices if all zones have at least one mobile node inside it and, as a result, a CH. Therefore, the existence of a path can be found using about 11,000 addition operations and it further reduces to about 2,800 additions in a square network. Performing binary multiplication using the connectivity matrix is not a costly process. For the above example, the operations can be executed in a few microseconds on a state-of-the-art processor. However, partial binary multiplication of T_R is sufficient in most cases.

The above process checks for path existence between the source-destination pair. The actual path can be constructed by inspecting the binary multiplication results in a backward manner, i.e. if the binary multiplication of T_R was performed i times and results in a value of 1, the actual path of i hops can be constructed by backtracking through $T_R^{i-1}, T_R^{i-2}, \ldots, T_R^1$. After the path discovery process

[9] If all links are symmetric, then the number of computations can be halved. However, we consider the general case of asymmetric links.

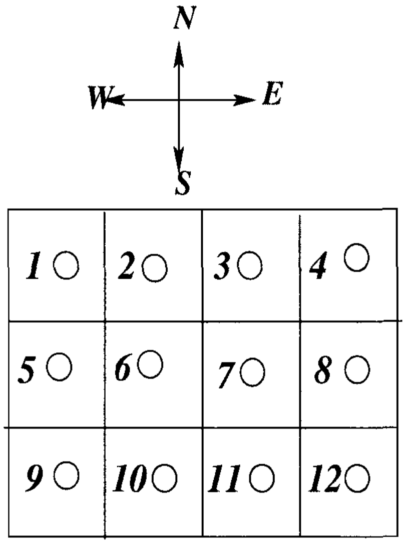


Figure 2.10 Routing over VGA using transitive closure approach.

$$T_R = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Figure 2.11 Adjacency matrix, T_R, of the example shown in Figure 2.10.

```
1 1 1 0 1 1 0 0 1 0 0 0    0 1 1 1 1 1 1 0 1 1 0 0    1 1 1 1 1 1 1 1 1 1 1 0    1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 0 0 1 0 0    1 0 1 1 1 1 1 1 1 1 1 0    1 1 1 1 1 1 1 1 1 1 1 1    1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 0 1 1 1 0 0 1 0    1 1 0 1 1 1 1 1 0 1 1 1    1 1 1 1 1 1 1 1 1 1 1 1    1 1 1 1 1 1 1 1 1 1 1 1
0 1 1 1 0 0 1 1 0 0 0 1    1 1 1 0 0 1 1 1 0 1 1 1    1 1 1 1 1 1 1 1 0 1 1 1    1 1 1 1 1 1 1 1 1 1 1 1
1 1 0 0 1 1 1 0 1 1 0 0    1 1 1 0 1 1 1 1 1 1 1 0    1 1 1 1 1 1 1 1 1 1 1 1    1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 0 1 1 1 1 1 1 1 0    1 1 1 1 1 1 1 1 1 1 1 1    1 1 1 1 1 1 1 1 1 1 1 1    1 1 1 1 1 1 1 1 1 1 1 1
0 1 1 1 1 1 1 1 0 1 1 1    1 1 1 1 1 1 1 1 1 1 1 1    1 1 1 1 1 1 1 1 1 1 1 1    1 1 1 1 1 1 1 1 1 1 1 1
0 0 1 1 0 1 1 1 0 0 1 1    0 1 1 1 1 0 1 1 0 1 1 1    1 1 1 1 1 1 1 1 1 1 1 1    1 1 1 1 1 1 1 1 1 1 1 1
1 0 0 0 1 1 0 0 1 1 1 0    1 1 0 0 1 1 1 0 1 1 1 1    1 1 1 0 1 1 1 1 1 1 1 1    1 1 1 1 1 1 1 1 1 1 1 1
0 1 0 0 1 1 1 0 1 1 1 1    1 1 0 0 1 1 1 0 1 1 1 1    1 1 1 1 1 1 1 1 1 1 1 1    1 1 1 1 1 1 1 1 1 1 1 1
0 0 1 0 0 1 1 1 1 1 1 1    1 1 1 0 1 1 1 1 1 1 1 1    1 1 1 1 1 1 1 1 1 1 1 1    1 1 1 1 1 1 1 1 1 1 1 1
0 0 0 1 0 0 1 1 0 1 1 1    0 0 1 1 0 1 1 1 1 1 1 0    0 1 1 1 1 1 1 1 1 1 1 1    1 1 1 1 1 1 1 1 1 1 1 1
```

Figure 2.12 Results for binary multiplication of boolean matrix T_R for connectivity with 2, 3, 4, and 5 hops, respectively.

ends in a successful finding of a path between the source and the destination nodes, the packet delivery process begins. The complete path directions is included in the packet header using the standard four directions. When a CH receives a packet from its neighbor CH, it will inspect the routing list (full path) embedded in the packet header, and forward the packet to the proper neighbor or direction accordingly. The following example illustrate the above procedure. Consider the virtual topology shown in Figure 2.10. The adjacency matrix for this (3 × 4) VGA is shown in Figure 2.11. Figure 2.12 shows the binary matrix multiplication results of matrix T_R for $n=$ 2, 3, 4, and 5 times (hops). Actual routes can be obtained by backtracking the multiplication process as was described earlier. For example, the backward inspection in this example returned the path 11-10-9-5-1 between the CH nodes 1 and 11 as a source and destination nodes, respectively. In this case, source node 1 will embed the following directions in the packet header (10101111) corresponding to the directions (EESS) and then send the packet to node 11. For this example, the existence of a path is found using about 256 addition operations.

2.4.3 Route Maintenance

There are two reasons for packet re-routing over VGA: broken links and congested links. In our scheme, a packet will reach the destination through the set of CHs that constitute the path between the source and destination pair (s, d). In the case of a CH failure, i.e., a zone becoming empty, the protocol should be able to route packets around the failed CH and continue serving packets. The responsibility of detecting whether the next CH has failed or not falls on the direct upstream CH in the packet route list, which is embedded in the packet header, and it is performed as follows. When a CH needs to forward a packet, it first checks the forwarding path (called *route-list*) to decide where to forward the packet next. If the link to the next hop is not available, e.g., a zone becomes empty or node failures, the path will fail. As a result, an error message can be sent back to the sender and the sender will initiate a new routing discovery process or use one of the backup paths found earlier.

However, trying local re-routing before the global re-routing may help alleviate the problem and provide quick recovery; hence improving the network performance. We refer to the re-routing scheme we use as (*local-then-global*). Depending on the situation, the current CH may find an alternate path on demand by contacting its neighboring CHs that may know a path to the destination. If no neighbor CH has a knowledge of a path, two things are done. First, the direct upstream CH will perform a quick route discovery since it has all the information about the destination node, and sends the packet once the path is computed using, for example, the transitive closure approach discussed earlier. Note that once the destination zone ID is known, computing the route from the source zone to the destination zone can be done in a few microseconds using the transitive closure approach, while finding the total route using on demand routing (RREQ-RREP) will take several milliseconds. Therefore, discovering new routes and re-routing is faster using the transitive closure approach, and hence this approach will be used in the route maintenance phase. Second, it informs the source node about original route failure and the rerouted path, such that source will decide whether to continue transmitting packets with this locally rerouted path or resort to another global solution, i.e. send packet on a different route where backup routes can be used.

In Figure 2.13, we illustrate the concept of local-then-global re-routing over VGA due to broken or congested links. The figure shows the current active path from zone 1 to zone 8 between a certain source-destination pair in these two zones, respectively, and its evolution with time as mobile nodes move. In Figure 2.13(i), the CH in zone 5, which is the only node in this zone, is moving towards

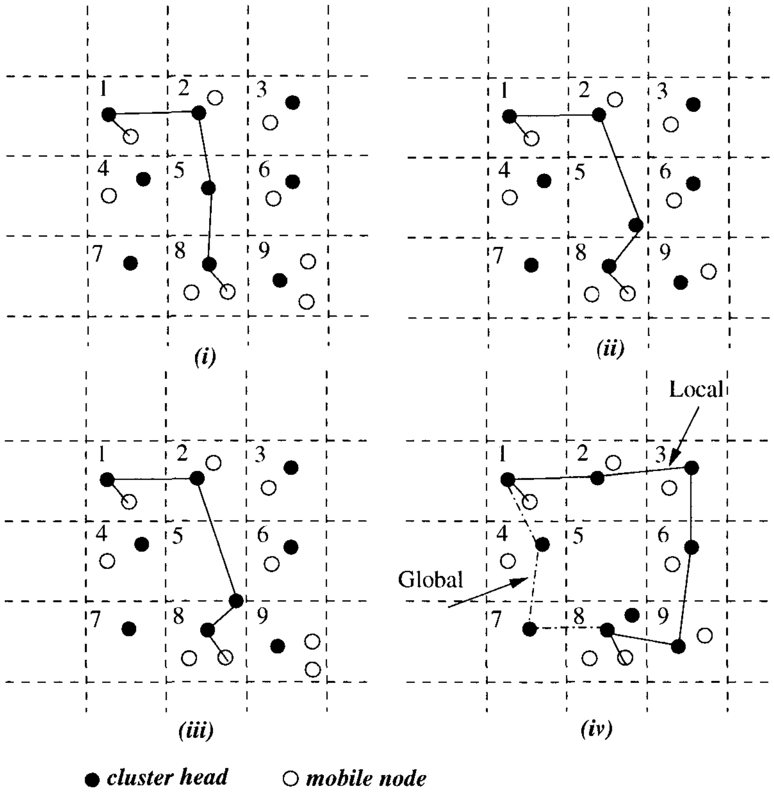


Figure 2.13 The illustration of local-then-global path restoration in VGA.

zone 9 while a traffic stream is passing through it. As the CH leaves its zone and moves to a different zone, service disruption should be prevented. One solution is that the moving CH notices that it is approaching another zone and announces that it is leaving, hence it informs the current previous CH to look for an alternate route. Early CH election in zone 5 will take place in parallel to this process. In Figure 2.13(*iv*), the moving CH is out of its zone and has entered zone 9, but the current CH in zone 2 was able to find an alternate path locally and continues the connection without service disruption. Thus, recovery from the interruption is achieved quickly by producing an alternate path locally, However, a global receeovery can be selected by the source node. As such, the source node can discontinue the current path and use an alternate global path (shown as dashed lines) in the Figure 2.13(*iv*) (going through zones 4, 7, and 8, respectively). The CH in zone 2 modifies the route list returned by the neighbor CH before sending the packet. The above discussion results in the reduced transmission delay of packets directed to a broken or congested link and the increased packet delivery ratio of the routing scheme.

2.5 Probabilistic Mobility Model for MANETs

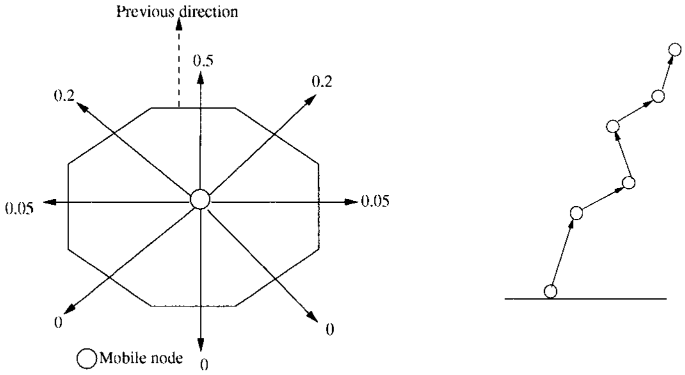
In Mobile Ad hoc Networks, mobile nodes roam around in the network area according to a certain mobility model. Traditionally, the simplified random Waypoint mobility model [19] is used to capture node mobility in the popular simulator used in MANETs, called Network Simulator NS-2 [50]. The random Waypoint mobility model includes pause times between changes in direction and/or speed. A mobile node begins by staying in one location for a certain period of time (i.e., a pause time). Once this time expires, the node chooses a random destination in the simulation area and a speed that is uniformly distributed between [minspeed, maxspeed]. The node then travels toward the newly chosen destination at the selected speed. Upon arrival, the MN pauses for a specified time period before starting the process again. It has been shown that the waypoint model has many performance issues that make it unrealistic [19]. In fact, each of the existing mobility models is based on certain intuitions and assumptions and might not correctly model realistic use of the system. There is no indication that there is a single-most correct mobility pattern that will accurately capture all the behavior of an ad hoc network. To achieve accurate user mobility prediction, the user mobility model needs to incorporate user behavioral patterns, wireless link characteristics, and the handover decision-making mechanism. Unfortunately, these are complicated problems. Hence, the problem of designing realistic and reproducible mobility models is hard.

Nevertheless, we now present a mobility model, called Probabilistic Mobility Model (PMM), that most closely represent the real world. Although this model guarantees that we do not stray too much from realistic world, it represents a general-purpose method that may be used to reliably generate realistic mobility patterns with different characteristics. We compare this model to a real user mobility traces collected in [135].

We assume that the initial speed of a mobile node is chosen to be a random variable with truncated Gaussian probability density function $f_V(v)$ having a mean and standard deviation of μ_v and σ_v, respectively. The choice of such a distribution seems to be reasonable since the more extreme the speed value, the less the likelihood of its occurrence. Due to speed limitations, it is also very unlikely that the speed exceeds a certain maximum value, v_{max}, with a typical value of 100 Km/h. The density $f_V(v)$ is given by:

$$f_V(v) = \frac{K}{\sigma_v\sqrt{2\pi}}e^{-(v-\mu_v)^2/2\sigma_v^2} \qquad 0 \leq v \leq v_{max}$$

and K is a normalization constant that depends on the values of μ_v, σ_v, and v_{max}. In the simulations, we vary the average initial speed of a mobile node, μ_v, as well as the standard deviation, σ_v in order to study the effect of node speed on the network performance. At any time instant, the mobile node speed is correlated with the previous speed, v_p. The current speed, v_c, of each mobile node is taken to be uniformly distributed random variable in the range of ±10% of the previous speed. Therefore, the



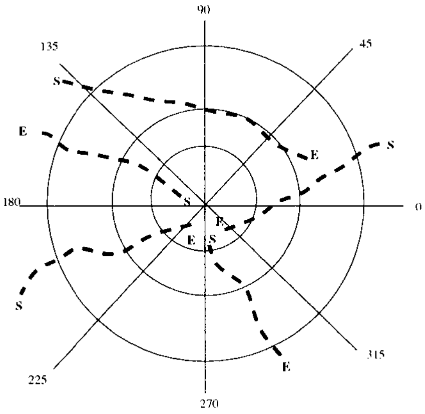


Figure 2.14 One-step Markov path model with memory and possible directions probabilities.

Figure 2.15 Movement in most probable directions, and path trajectory trace of five mobile nodes (S and E: Start and End points).

mobile speed increment pdf, $f_{V_c}(v_c)$ will be:

$$f_{V_c}(v_c) = \begin{cases} \frac{1}{0.2v_p} & , 0.9v_p \leq v_c \leq 1.1v_p \\ 0 & \text{Otherwise} \end{cases}$$

As for the node's movement direction, we assume that the mobility model has a memory as was assumed in the one-step Markov path model [28]. In the Markov path model, mobile nodes are allowed to move away from the starting point in any direction with equal probability. After that, a mobile node has a higher probability of moving in the same direction as the previous move. However, the mobile node direction is biased towards the direction of its destination. Figure 2.14 shows the probability assignment in eight different directions as used in our simulations. Note that the probability of reversing moving direction is set to 0. As the moving direction and the speed of mobile nodes are non-deterministic processes, the path of a mobile node will be a random trajectory. Figure 2.15 shows the movement in most probable directions of a mobile node. It also shows a sample trajectory paths collected from simulation experiments for five users wandering in the network area with starting points (S) and ending points (E). The path trajectories resemble a smooth movement of each user, which may represent a more realistic movement pattern. The changes in direction occur in time steps that are exponentially distributed with an average value of one minute. The motivation for the choice of an exponential distribution for the length of time between two consequent changes of direction because the time of the last change in direction hardly provides any information about the time of the next change in direction.

Performance of PMM: Most researchers propose customized mobility models and validate their protocols against these models. If the models are unrealistic, invalid conclusions may be drawn. Due to the variety of models, it is impossible to easily compare performance results of two protocols. SUMATRA

[135] is a trace generator that is well-validated against real data on mobility traces. In this part, we compare the performance of PMM model to a real trace scenario taken from [135] in terms of mobility patterns. The simulation scenario is similar to that in [135]. The network area is square grid that was divided into 36 zones (6 x 6) with an average of 36 mobile nodes per zone. The trace duration is 4 hours. If the mobile node reaches the border of the network area, it appears again from the opposite border (torus-like movement). Start points of mobile nodes were randomly generated. The real trace scenario file has a size of about 140 MB. In Figure 2.16, we plot the real trace mobility pattern for two users wandering in the grid. In the same figure, we compare the mobility pattern of two users under PMM model. The mobility shows that PMM model is capturing the same mobility pattern of the two users real traces. In fact, we experimented with more number of users. However, we are not showing the mobility patterns of these users in order not to clutter the figure, but they showed the same trend of mobility pattern as the real traces. This indicates that PMM model has the potential to be a good mobility model for MANETs. The question whether a fine tuning of PMM parameters will enhance its performance is still under investigation.

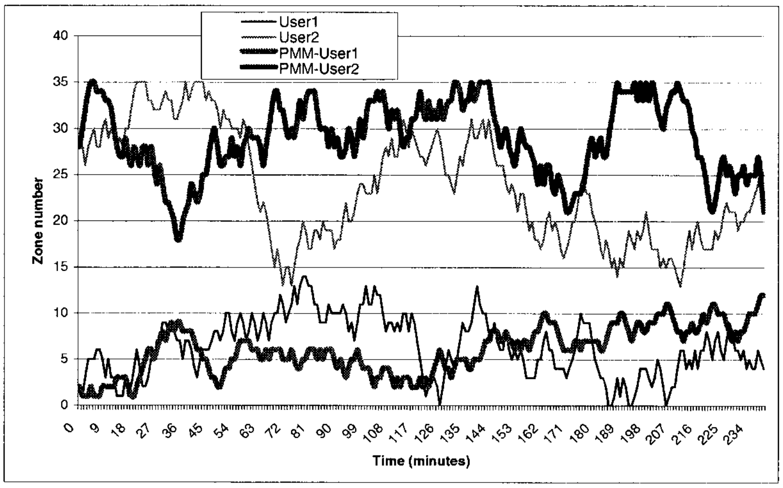


Figure 2.16 Mobility pattern of users under PMM model and real trace scenarios.

2.6 Simulation Results

In this section, we present the results of applying the routing protocols proposed in this chapter in both homogeneous and heterogeneous networks with different parameters. The two routing techniques proposed in Section 2.4 were simulated using the NS 2.26 simulator [50]. A heterogeneous MANET in a fixed area of size (2000m×2000m) with a variable number of mobile nodes was simulated. Unless mentioned specifically, the number of mobile nodes in the simulation are 500. Mobile nodes were initially placed randomly within the fixed-size network area. The long range transmission distance is set to 250 meters and short range transmission distance is set to 125 meters. Links have a transmission rate that is equiprobably selected from a pre-defined set of transmission rates (1, 2, 5.5, and 11 Mbps). When a new session is generated by a mobile node, the routing protocol is initiated. Connections are established between randomly selected pairs of source/destination nodes. For the communication pattern, we used bursty traffic with ON-OFF periods. The ON and OFF periods of each flow are exponentially distributed with mean values being simulation parameters. During ON periods, traffic is generated at a constant rate of 20packets/sec, and then sent between source-destination pairs selected from a uniform random distribution of all nodes in the network. During OFF period, the source does not generate packets and remains idle. The traffic flows were assumed independent. New session requests arrive according to a Poisson arrival process with mean arrival rate of 0.1. The length of each flow session is exponentially distributed with the mean value being a simulation parameter. We vary the session duration, the ON period length of each session, the packet generation rate during the ON period, and the mean OFF period length to obtain different levels of the offered load. Unless otherwise specified, the default offered traffic load was set to 0.8. Multiple communication sessions from a source node or toward a destination node are not allowed. Each simulation experiment runs for a duration of 5000 seconds. No data collection was performed for the first 100 seconds of the simulation to avoid transient period and to ensure that the initial route discovery process is stable. The packet sizes are fixed at 512 bytes. The mobility model used is PMM, which was presented in Section 2.5.

In the CH election algorithm, the weighting parameters were selected as follows: $a_i = 0.25, i = 1 \ldots 4$. In the on-demand CH election scenario, a CH node will initiate a new, and possibly, early CH election when its battery capacity drops down to 20% of its initial capacity. Since it is impossible to evaluate the behavior of the network if the nodes generating traffic run out of energy before the end of simulations, we give nodes infinite energy. In another setting, we limited the initial node energy to study the effect of limited energy budget on the CH election algorithms and on the network lifetime. For simplicity, we assume that for each packet, an amount of energy from the battery is consumed as follows:

$$E(packet) = a * (b * (packet\_size) + c)$$

where a, b, and c are weighting parameters, a is an operation dependent parameter, b is packet size dependent parameter, whereas c is a fixed cost that accounts for acquiring the channel and for MAC layer control negotiation. The parameters of the above equation are set as follows: a is set to 1.5 for receiving, 3.0 for transmitting, and 1.0 for listening. b is set to 0.5 mW/byte and c to 0.1 mW. We evaluate the performance of the proposed routing techniques based on the following metrics: (a) Packet delivery ratio (b) Average end-to-end packet delay (c) Route breakage (d) Call acceptance ratio (e) Routing Overhead (f) Clusterhead election period, (g) Normalized power consumption, (h) Scalability, and (i) Node cooperation.

(a) Packet Delivery Ratio: We define the packet delivery ratio as the ratio between the number

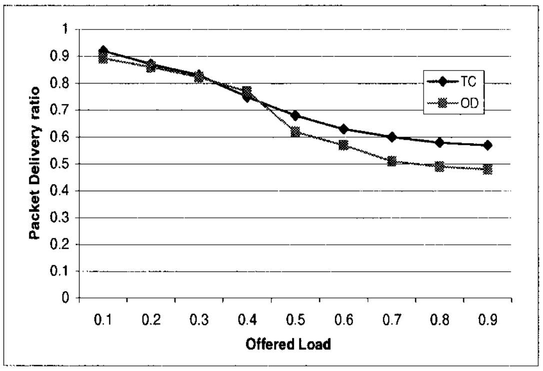


Figure 2.17 Packet Delivery Ratio vs. offered load.

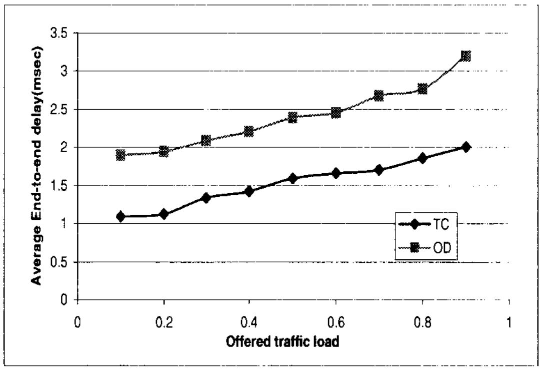


Figure 2.18 Network end-to-end delay vs. offered load.

of packets received by the destination and the number of packets generated by the application layer sources of the accepted calls. Packet delivery ratio is important as it describes the loss rate that will be seen by the transport protocols, which in turn affects the maximum throughput that the network can support. Figure 2.17 shows the percentage of the packets delivered using both on-demand routing (OD) and transitive closure (TC) over VGA. Both schemes are able to achieve high packet delivery ratios and they maintain acceptable levels even when the offered load in the network increases. Stability of VGA allows both OD and TC based routing techniques to achieve high packet delivery. However, TC-based routing technique outperforms OD-based routing at higher offered load. This is because TC-based routing is able to locate paths quicker than OD-based based on local information and computations.

Recall that the optimal path is defined as the path with the minimum number of hops since it uses the fewest resources. We measured the percentage use of optimal paths used in TC-based routing. It is found that TC-based routing is able to use around 82% of the optimal paths when serving various calls. This efficient use of the resources also explains why VGA-based routing techniques were able to achieve high packet delivery ratio.

(b) Average end-to-end packet delay: The average end-to-end packet delay versus the offered traffic load for both routing techniques is shown in Figure 2.18. Note that a link becomes unreliable when it is broken and/or saturated with heavy traffic. When a link is unreliable, the node fails to forward packets, causing packet drops or longer delays. At low traffic load, nodes rarely experience congestion but often experience broken links. Therefore, packets that need to be re-routed will be queued and therefore encounter longer delays. Both routing techniques employ the efficient re-routing solution (*local-then-global*) discussed previously in order to reduce the average end-to-end packet delay. In Figure 2.18, it can be observed that TC-based routing improves the average end-to-end delay by as much as 20% over OD-based routing. At high traffic load levels, however, the nodes drop packets despite the presence of paths since the forwarding rate is far lower than the packet arrival rate. However, both

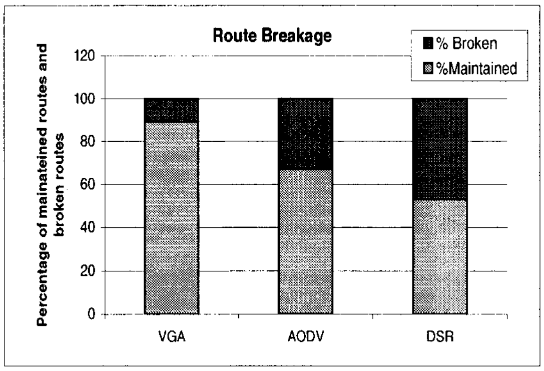


Figure 2.19 Percentage of broken routes due to mobility.

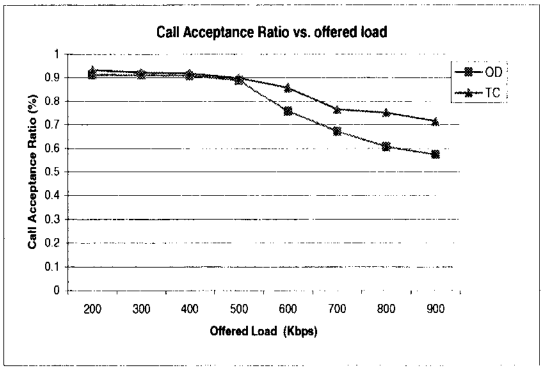


Figure 2.20 Call acceptance rate in the two routing techniques.

protocols attempt to relieve congestion by finding an alternate path and re-route the traffic.

(c) Route Breakage: We measured the percentage of broken routes over VGA when the OD-based routing is used and compared the results to the well-known routing protocols, AODV and DSR. The mobility model used is PMM. Figure 2.19 shows this comparison. It is obvious from the figure that OD-based routing running on top of VGA is able to maintain more than 87% of the discovered routes, while the traditional AODV was able to maintain less than 65% of the routes while performing better than DSR.

(d) Call acceptance ratio: Call acceptance ratio is defined as the number of successfully accepted route requests divided by the total number of requests generated in the network for the length of the simulation period. Calls are rejected if routes to the required destinations cannot be found. Figure 2.20 shows the call acceptance ratio as a function of the offered traffic load for both routing criteria. As the figure shows, both the OD- and TC-based routing techniques are able to achieve more than 90% call success ratio at light loads. However, TC-based routing technique shows a better performance than OD-based routing as the offered traffic load increases. This is because the distributed OD-based routing technique took longer to find a path compared to the TC-based approach.

(e) Routing Overhead: This is defined as the number of routing packets per second as measured over the entire simulation experiment. Each transmission of a routing packet over each hop counts as one transmission. Routing overhead is an important metric for comparing routing protocols, as it measures the scalability of a protocol, the degree at which it will function in congested or low-bandwidth environments, and its efficiency in terms of consuming battery power. Protocols that send large numbers of routing packets can also increase the probability of packet collisions and may delay data packets in the network queues. Figure 2.21 shows the routing overhead of the two routing techniques for variable offered traffic loads compared to the dynamic source routing (DSR) protocol. It can be seen that both routing techniques running over VGA are able to use lower control overhead than DSR. Moreover, TC-based routing requires lower control overhead than OD-based routing, especially as the offered load increases.

(f) Cluster head election strategies: We also studied the effect of having fixed or variable periodic CH election as well as having on-demand CH election on the performance of the two routing techniques running on top of VGA. We measured the packet delivery ratio as the offered load increases. We tried different fixed values for the CH election period and the reported results are for a fixed period of 2 seconds. For a variable period CH election, we developed a simple formula to set the CH election period in each zone such that the length of this period is dependent on three parameters at the beginning of CH election process: the number of nodes in the zone (n), the elected CH node available energy (e), and the speed of the elected CH (v). CH election is executed in each zone whenever the CH election timer expires. Note that due to node mobility, membership of each zone changes and this affects the CH election algorithm. One simple equation is therefore: $CH_{period} = \frac{e}{v*n}$. Naturally, the CH period in the network decreases when there are more nodes in the zone or when the current CH moves quickly. As for the on-demand CH election strategy, a node can demand a new CH election if its energy level drops below 50% of its initial energy capacity. For this purpose, we assigned each mobile node a fixed amount of initial energy capacity of 8000J. Figure 2.22 shows the difference between the three CH election strategies when the CH election period is fixed to 2 seconds and when the CH election period is variable and is computed in each zone using the previous equation. For this experiment, each mobile node velocity v is initially selected per the PMM model such that it falls between (0~5 m/s). We fixed the number of nodes in the network to 100. As can be seen in the figure, there is a tradeoff between the three strategies for different traffic loads. For low to moderate traffic loads, fixed-period strategy resulted in the higher packet delivery ratios. However, on-demand strategy performs better at higher offered loads which indicates that CH nodes were suffering and hence needed to be replaced more often. This resulted in more load balancing among mobile nodes. A hybrid approach may solve this problem at the expense of higher implementation costs.

(g) Normalized energy consumption: This is defined as the total energy consumed in delivering the packets of accepted calls. We measure the energy consumption because energy is a precious resource in mobile communications. Figure 2.23 shows the normalized energy consumption incurred in OD-based routing working on top of VGA. The energy consumption increases gradually with increasing the offered load, but maintains acceptable levels of energy consumption at higher loads. As a result, the network (node) lifetime is prolonged. This is a result of the use of the simple and efficient power control scheme and the stability of the virtual topology VGA against node mobility. To further solidify this observation, we measured the difference between the minimum and the maximum node lifetime under OD-based routing technique compared to plain AODV. Figure 2.24 shows this comparison. OD-based routing technique that works on top of VGA is able to extend node lifetime more than AODV. This

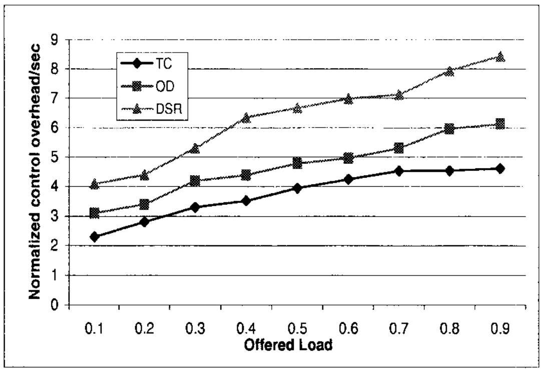


Figure 2.21 Network control overhead versus offered load using the TC and OD techniques.

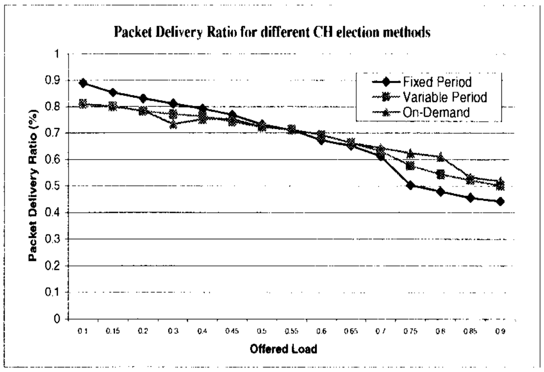


Figure 2.22 Packet delivery ratio versus offered load under different CH election strategies.

is due to the use of the power control scheme of VGA. The bars shown in the figure represent the difference between minimum and maximum node life time for each offered load. Note that VGA is able to balance the energy consumption among mobile nodes which results in extending their lifetimes.

(h) Scalability: Scalability is an important factor that refers to the adaptability of the protocol to larger networks, in terms of area and number of nodes. We use the same simulation environment presented previously except for the number of nodes and the grid (network area) size. We increase the network size to 3000m×2000m, and vary the number of nodes between 100 to 500. We generated 30 CBR sessions for this experiment. Each node moves with a speed randomly generated between 2~20 m/s using the PMM model. Figure 2.25 shows the scalability of VGA clustering approach. As the the number of nodes increases, the normalized control overhead of OD-routing over VGA scales almost linearly with the increment in the number of mobile nodes. Furthermore, as the number of nodes increases, the overhead tends to decrease due to the stability of VGA, which is enhanced by the increase in node density in the network. When the node density is low meaning that we have sparse node distribution, the number of available paths was limited and hence the control overhead is also limited.

(i) Enforcing node cooperation: In this part, we study the performance of the "enforcing node cooperation" scheme presented earlier in this Chapter. Our interest is to study the effect of user selfish behavior on the CH election algorithm in terms of the per node packet delivery ratio and the per node average energy consumption in the network. We are also interested in studying the effect of varying the number of selfish users on the call acceptance rate in the network. We modified the CH election algorithm to take into consideration the effect of selfish behavior as per equations (2.3) and (2.4). The value of w_i^p of a node i was selected based on the fraction of available energy at node i, E_i, at the beginning of the election period, i.e., $w_i^p = E_i$. First, we measure the average energy consumption per

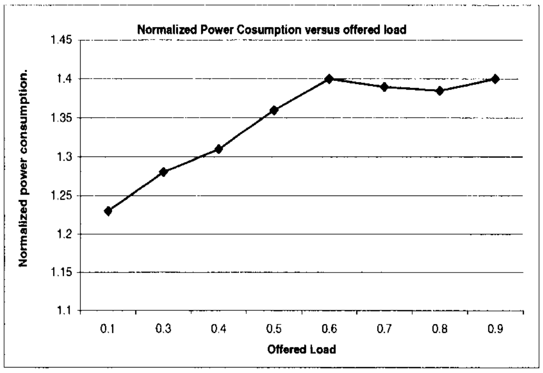


Figure 2.23 Normalized power consumption as load increases.

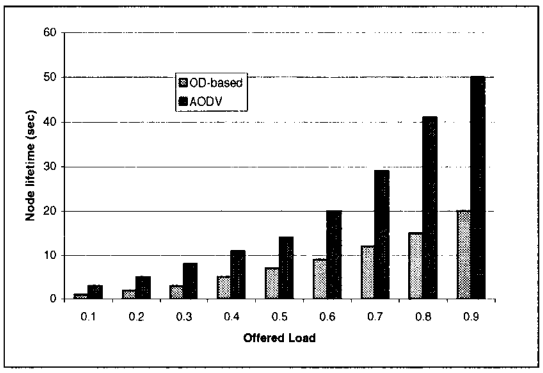


Figure 2.24 Effect of power control scheme of VGA on node lifetime.

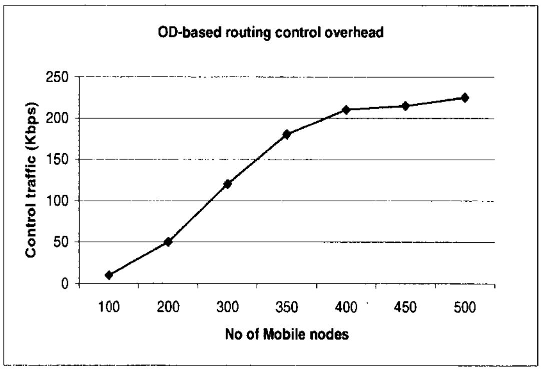


Figure 2.25 Od-based routing scalability as node density increases.

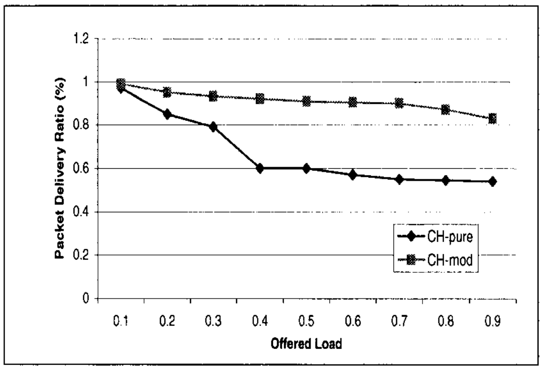


Figure 2.26 Effect of selfish behavior on packet delivery ratio.

node in the intra-zone cooperation. We fix the number of nodes to 350. Our CH election is the periodic strategy with fixed period set to 3 sec. Figure 2.26 shows the packet delivery ratio when the offered load changes when both the original CH election algorithm (CH-pure) and the CH election algorithm with the effect of selfish behavior is added (CH-mod). It is noted from the figure that the scheme is able to enhance the packet delivery ratio by enforcing nodes to cooperate for the benefit of all. We also measured the node power consumption variance for both cases as shown in Figure 2.27. The addition of the scheme shows that average node power consumption decreased, which helps prolong the network operational lifetime.

Then, we study the effect of users behavior on the blocking probability, that is the number of rejected calls, for the inter-zone cooperation. We assume that two types of users exist. The first type is cooperative users who apply the same strategy but with different values of willingness, while the second type is selfish users with low willingness values. In particular, we consider a group of cooperative users with willingness randomly generated between 0.8 and 1.0 and a group of selfish users with willingness

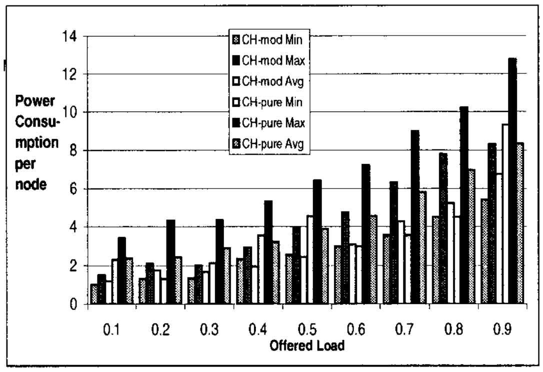


Figure 2.27 Effect of selfish behavior on the node power consumption.

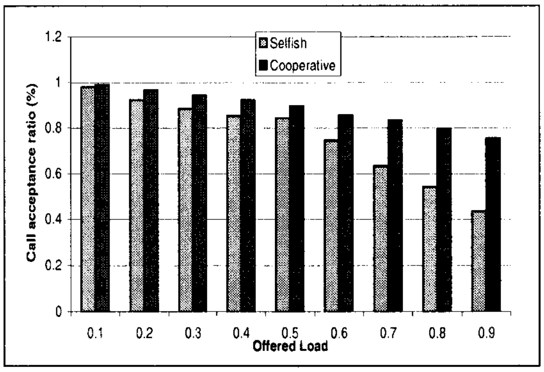


Figure 2.28 Effect of selfish behavior on call acceptance ratios for variable offered load.

equal to 0.5. This scenario may correspond to the case where users have different energy constraints or the case where some users are acting selfishly to maximize their own benefit. We fixed the total number of users to 100. We used the TC-based routing scheme. We measured the percentage of accepted calls for each user type when the offered load is varied. Figure 2.28 shows the call acceptance ratio for various values of offered load for both selfish and cooperative users when the ratio of selfish to cooperative users is (0.3,0.7). At low loads, the system is able to support both users requests. As the load increases, the system is able to differentiate between the two user types and favor the cooperative users.

We then varied the number of users in each type such that the user ratios range from (0.1,0.9) to (0.9,0.1) corresponding to the fraction of selfish and cooperative users in the network, respectively, and measured the call acceptance rate in the system for both the original routing scheme and the modified one. In Figure 2.29, the curves labeled by "cooperative" and "selfish" represent the performance of cooperative and selfish users, respectively under the original routing scheme and the modified routing with the inter-zone cooperations scheme. As the number of selfish users increases, the number of calls accepted for them decreases, while cooperative users managed to get over 70% of their calls through. This, reflects that the system is able to prevent selfish users from making advantage of cooperative users.

2.7 Chapter Summary

In this chapter, we addressed the important issue of topology control in both homogeneous and heterogeneous ad hoc networks by developing a simple, fixed, and scalable virtual wireless backbone. The virtual backbone, called Virtual Grid Architecture (VGA), is created through a novel and simple tessellation (zoning) scheme with low overhead. The tessellation scheme maps the network physical topology onto a virtual (*rectilinear*) topology. VGA consists of a few, but possibly more powerful,

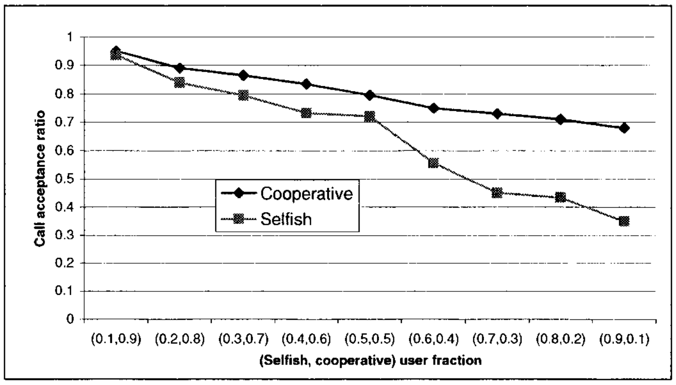


Figure 2.29 Effect of selfish behavior on the call acceptance rate for various populations.

nodes known as *ClusterHeads* (CHs) that are selected based on several strategies. A simple power control scheme, which captures differences in node transmission power, is used in conjunction with the zoning process to ensure network connectivity while saving power consumption. We also presented a scheme for stimulating nodes to cooperate in the CH election and routing process in VGA. The virtual wireless backbone, VGA, simplifies the routing function in infrastructerless wireless networks. We have presented two routing techniques, namely, On-Demand based routing and transitive closure based routing. An extensive performance study of several aspects of the proposed protocols was conducted. Both routing techniques, working over VGA, showed an enhanced performance in terms of number of accepted calls, packet delivery ratio, routing overhead, and energy consumption. It can be concluded that stable and fixed routing architecture can enhance the performance of MANETs in terms of packet delivery and call acceptance ratios.

CHAPTER 3 Properties of VGA Clustering Approach

In the previous chapter, we have presented the VGA clustering approach and presented two routing techniques that use VGA as the underlying routing structure in MANETs. In this chapter, we study the properties of the VGA clustering approach. In particular, we focus on different and important aspects of VGA, e.g., VGA size, route length over VGA, clustering overhead, connectivity, and stability.

3.1 VGA Clustering: Simplicity versus Optimality Tradeoffs

In this section, we study the performance tradeoffs between the proposed simple VGA clustering approach and an optimal clustering approach for both homogeneous and heterogeneous Mobile Ad hoc Networks (MANETs). First, we justify the use of rectilinear virtual topology. Second, we find the CHs cardinality, the worst case and average case path length in VGA as well as the control overhead of the VGA clustering approach. Finally, we analyze the connectivity and stability of the proposed VGA clustering approach.

3.1.1 The Homogeneous Network Case

In this section, we study the performance of the proposed clustering approach and compare it to optimal clustering in homogeneous MANETs. First, we analytically justify the use of the rectilinear virtual topology and explain why diagonal routing (referred to as Diagonal VGA (D-VGA)) is not adopted as a virtual topology. We then find closed form expressions for the CHs cardinality, and the worst case and average case path lengths in terms of the number of hops in VGA. Figure 3.1 shows the zoning process and the resulting virtual topologies of VGA, D-VGA, and optimal zoning, respectively. The optimal network coverage and the least number of fixed zones are achieved when the network area is divided into fixed and equal *hexagons*, such that at the center of each hexagon there is a node acting as a clusterhead. This is very restrictive and can only be guaranteed with a very large number of nodes, which are uniformly distributed over the network area. The likelihood of this happening in MANETs is small, but will be used here for reference.

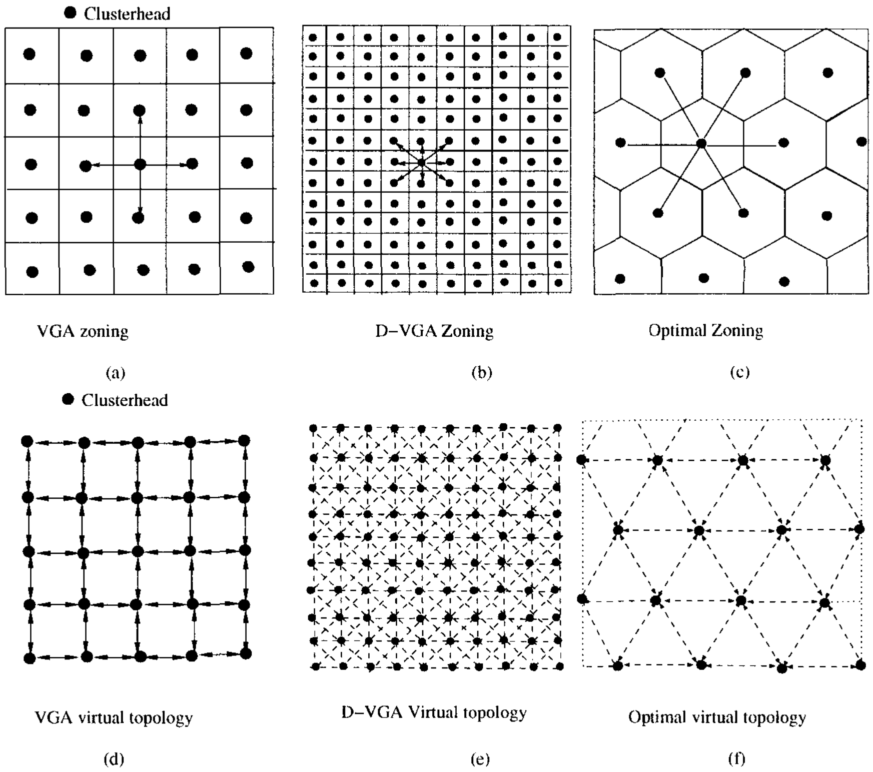


Figure 3.1 Zoning in (a) VGA, (b) D-VGA, and the (c) optimal zoning. The virtual topology corresponding to the above tessellations is shown in (d) VGA, (e) D-VGA, and (f) Optimal cases.

3.1.1.1 Rectilinear Routing versus Diagonal Routing

In VGA, a CH communicates only with its vertical and horizontal neighbors directly. In D-VGA, however, a CH communicates with all of its eight neighbors. The number of zones in D-VGA is almost twice that of VGA per network area as zone size is smaller in D-VGA. To illustrate, let the network area $A = L \times L$. In VGA, the zone side length is $x_v = \frac{r}{\sqrt{5}}$, while the zone side length of D-VGA, $x_d = \frac{r}{\sqrt{8}}$. Hence, the number of zones in VGA is $\frac{L}{x_v} * \frac{L}{x_v} = \frac{5L^2}{r^2}$, while it is $\frac{L}{x_d} * \frac{L}{x_d} = \frac{8L^2}{r^2}$ in D-VGA. This represents an increase in the number of zones in D-VGA by a factor of 8/5. Consequently, the average number of routing hops increases in D-VGA, as will be shown later in this chapter. Since packets have to be relayed by a larger number of CHs in D-VGA, then higher latency, more total power consumption, and more information processing per node will be incurred when diagonal routing is used. To illustrate this point, we now show that the probability of diagonal routing succeeding given that rectilinear routing has failed, is low.

Consider a target zone, and its eight neighboring zones as shown in Figure 3.2(a). Let B be the event where the target zone has at least one node, the neighboring vertical and horizontal zones are

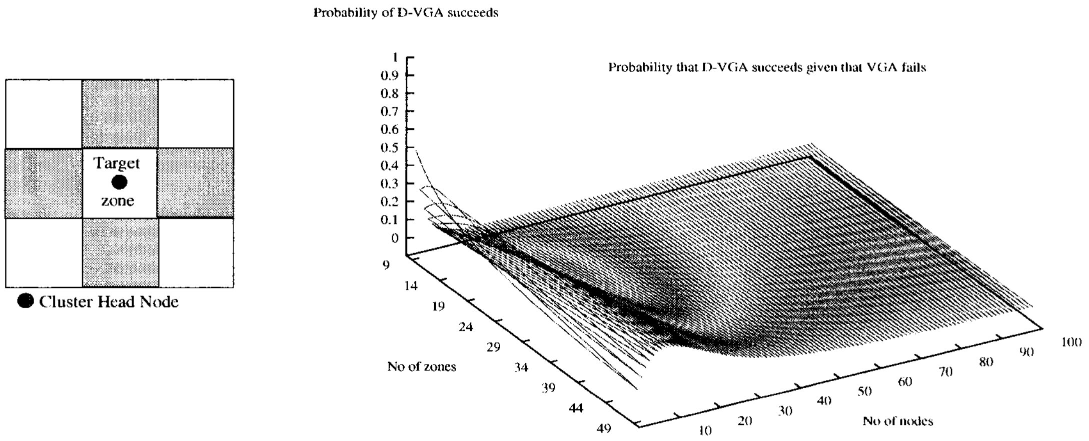


Figure 3.2 (a) Diagonal routing capability option (b) Probability that D-VGA succeeds given that VGA fails.

empty, and at least one diagonal zone is not empty. Also, assume that the total number of nodes in the network area is N, and that there are $|Z|$ zones in the network area. Let $P(B)$ be the probability of the occurrence of event B (i.e., the rectilinear routing from the central zone fails while the diagonal routing succeeds). We express $P(B)$ as:

$$Pr(B) = \sum_{n=1}^{N} Pr(B|n)\binom{N}{n}(\frac{|Z|-9}{|Z|-1})^{N-n}(\frac{8}{|Z|-1})^{n} \tag{3.1}$$

where $Pr(B|n)$ is the probability of the event B given that the n nodes are distributed uniformly in the four diagonal zones and is given by $\frac{\binom{n+3}{3}}{\binom{n+7}{7}}$. The expression $\binom{N}{n}(\frac{|Z|-9}{|Z|-1})^{N-n}(\frac{8}{|Z|-1})^{n}$ represents the probability that out of the total N nodes in the network area, n nodes are in the eight zones around the target zone (central zone). We plot $P(B)$ for different numbers of zones and mobile nodes in Figure 3.2(b). As shown in the figure, the probability of successful diagonal routing and rectilinear routing failure is very low. For example, when the number of zones $|Z|$=40 and the number of nodes N=40 the value of $P(B)$=0.035729. Hence, adding diagonal routing does not significantly improve the probability of finding a route in case rectilinear routing fails in addition to increased complexity and overhead associated with D-VGA.

3.1.1.2 Clusterheads Cardinality

We define *the cardinality* of the virtual topology, VGA, as the number of elected CHs. The CHs cardinality[1], $C(t)$, of the virtual topology at time t can be expressed as:

$$C(t) = \sum_{j=1}^{N} Y_j(t)$$

We find C for VGA, D-VGA, and for optimal clustering as follows. It is known that the area, R, of a hexagon (which is used in optimal clustering) with side length x is given by $R = \frac{3\sqrt{3}}{2}x^2$. Two CHs located at the centers of two neighboring hexagons will be able to communicate when $x = \frac{r}{\sqrt{3}}$; hence the area of the hexagon can also be written as $R = \frac{\sqrt{3}}{2}r^2$. For VGA and D-VGA, the zone sizes are given by, $R = x^2 = (\frac{r}{\sqrt{5}})^2 = \frac{r^2}{5}$ and $R = x^2 = (\frac{r}{2\sqrt{2}})^2 = \frac{r^2}{8}$, respectively. Assuming a large network area of size $L \times L$ square meters, the number of clusters C, (which is also the expected CHs cardinality) can be found easily as:

$$\text{VGA: } C = \frac{5L^2}{r^2}\text{ , D-VGA: } C = \frac{8L^2}{r^2}\text{, Optimal: } C = \frac{2L^2}{\sqrt{3}r^2}$$

Table 3.1 summarizes the results and also shows the ratio, defined as the number of clusters in VGA and D-VGA divided by the optimal number of clusters. We note from the table that VGA encounters less CH cardinality than D-VGA and it is close to the optimal CH cardinality. Note that the case of hexagons is unrealistic since we assumed for theoretical reasons that at the center of each hexagon there is a node acting as a clusterhead. This requires nodes to be static. However, if the worst case scenario for communication is considered in optimal (hexagon) zoning, then the ratio of VGA to optimal zoning would be smaller.

Table 3.1 Comparison of clusterhead cardinality for VGA, D-VGA, and optimal clustering.

Case	Optimal	VGA	D-VGA
Number of clusters(S)	$\frac{2L^2}{\sqrt{3}r^2}$	$\frac{5L^2}{r^2}$	$\frac{8L^2}{r^2}$
Ratio	1.0	4.33	6.93

3.1.1.3 The Worst-case and Average-case Path Length

The worst case path is defined as the path with the largest number of CHs between any two nodes, while the average case path is defined as the sum of lengths (number of hops) of all available paths from a source to a destination divided by the number of available paths. We now show that the average path length of VGA is shorter than the average path length of D-VGA. Also, we show that although VGA

[1] Onwards, we drop the time index for convenience, i.e., we use C instead of $C(t)$)

is simple, the average path length over VGA is closer to the optimal average path length than D-VGA. On the other hand, the worst case path length of VGA is higher than that of D-VGA. We state these facts formally in the following proposition (the proof is given in Appendix 1).

Proposition 1: If the network area is large and is equal to $L \times L$ square meters, mobile nodes are uniformly distributed in the network area, and each node has a transmission range of r meters, then the worst case path length (PL) of VGA, D-VGA, and Optimal routing (in terms of the number of hops) is given by:

$$PL_{VGA}^{max} = \frac{2L\sqrt{5}}{r} - 1 \ , \ PL_{D-VGA}^{max} = \frac{L\sqrt{8}}{r} \ , \ PL_{Opt}^{max} = \frac{L}{r}(\frac{1+\sqrt{3}}{\sqrt{3}}),$$

while the average case path length is given by:

$$PL_{VGA}^{avg} = \frac{\sqrt{5}L}{r} \ , \ PL_{D-VGA}^{avg} = \frac{5\sqrt{2}L}{3r} \ , \ PL_{Opt}^{avg} = \frac{(3\sqrt{3}+11)L}{18r}.$$

Proof. see Appendix A. □

The increase in D-VGA average path length can be intuitively justified by observing that in D-VGA, zones will be smaller in size. As such, the same network area in D-VGA will contain more zones than VGA, which leads to longer routes in terms of the number of hops. The results of our simulations and a simulation study carried in [29] are in agreement with our derived equations where the simulation results show that the average path length of VGA is smaller than D-VGA. Notice that this has a direct consequence of an increased control overhead and resource consumption under D-VGA due to its increased route length.

3.1.2 The Heterogeneous Network Case

In order to reduce the communication overhead and simplify the connectivity management, it is desirable to find a minimum connected virtual backbone. If the nodes in the virtual backbone communicate directly with each other, then finding the minimum connected virtual backbone is similar to finding the Minimum Connected Dominating Set (MCDS) of a given set of nodes in the network. However, finding MCDS is an NP-hard problem [52]. In this section, we provide an optimal solution for the problem of finding the set of connected CHs, i.e., finding MCDS in a group of mobile nodes forming an arbitrary connected graph. We formulate this problem as an Integer Linear Program (ILP). The proposed ILP works for both homogeneous as well as heterogeneous MANETs of small to medium size networks. First, we will outline the ILP for homogeneous MANETs and then show how the ILP can be easily extended to handle the heterogeneous case. Using the notation in Table 3.2, the optimal CHs selection problem (i.e., finding the MCDS) can be found by solving the ILP shown in Table 3.3. The objective of the ILP is to minimize the total number of CHs in the network.

Constraints (3.2) and (3.3) are for non-CHs to CH connections where constraint (3.2) guarantees that node i has a one hop connection to at least one CH, and constraint (3.3) guarantees that node i

Table 3.2 The ILP variables in heterogeneous MANETs.

Notation	Meaning
N	The number of mobile nodes in A.
c_{ij}	Adjacency indicator, which is 1 if and only if there is an edge between i and j in G; otherwise, c_{ij} is 0. Note that $c_{ij} \iff c_{ji}$. The values of c_{ij} are an input to the ILP.
y_{ij}	A binary indicator, which is 1 if and only if mobile node i uses j as its clusterhead. Note that y_{jj} is always 1 if and only if j is a clusterhead.
x_{ij}	A binary indicator, which is 1 if and only if both mobile nodes i and j are CHs.
r^h_{ij}	A binary indicator, which is 1 if a cluster head i can reach clusterhead j in h hops.

Table 3.3 The ILP formulation for finding MCDS.

Objective function: *Minimize* : $\sum_i y_{ii}$ (Number of CHs in G).

Subject to:

$$\sum_j y_{ij} \geq 1 \text{ , } \forall i \tag{3.2}$$

$$y_{ij} - y_{jj}c_{ij} \leq 0 \quad \forall i,j \ (i \neq j) \tag{3.3}$$

$$x_{ij} \leq \frac{y_{ii} + y_{jj}}{2} \quad \forall i,j \ (i \neq j) \tag{3.4}$$

$$x_{ij} \geq (y_{ii} + y_{jj} - 1) \quad \forall i,j \ (i \neq j) \tag{3.5}$$

$$r^1_{ij} = x_{ij}c_{ij} \ \forall i,j \ (i \neq j) \tag{3.6}$$

$$r^h_{ij} = \bigvee_k (r^{h-1}_{ik} \wedge r^1_{kj}), \forall i,j,h \ (i \neq j,k, k \neq j), 2 \leq h \leq N-1 \tag{3.7}$$

$$\sum_{h=1}^{N-1} r^h_{ij} \geq x_{ij} \ ; \forall i,j \ (i \neq j) \tag{3.8}$$

uses mobile node j as its CH. Constraints (3.4)-(3.8) are used for virtual backbone connections. The two constraints in (3.4) and (3.5) guarantee that if x_{ij}=1, then nodes i and j are both CHs, i.e., (x_{ij} = 1) $\iff$ (y_{ii}=1) $\wedge$ (y_{jj}=1). The last three constraints (3.6)-(3.8) guarantee that a clusterhead i is reachable by any other clusterhead j through at most h hops, $1 \leq h < N$, i.e., all CHs make a minimal connected dominating set where constraints (3.7) recursively find a route between two CHs i and j using other CHs as intermediate nodes, while constraint (3.8) makes sure that only intermediate CHs that are connected are used to find a route between CHs i and j.

For the sake of completeness, it should be mentioned that one can implement the conjunction, say

$X_1 = \bigwedge_{i=1}^{N} A_i$, $X_1 \in \{0,1\}$, and disjunction, say $X_2 = \bigvee_{i=1}^{N} A_i$, $X_2 \in \{0,1\}$, operations by using the following two constraints, respectively.

$$\underbrace{X_1 \leq \sum_{i=1}^{N} \frac{A_i}{N}, \quad X_1 \geq \sum_{i=1}^{N} A_i - N + 1}_{conjunction} \qquad \underbrace{X_2 \leq \sum_{i=1}^{N} A_i, \quad X_2 \geq \sum_{i=1}^{N} \frac{A_i}{N}}_{disjunction} \tag{3.9}$$

The solution of the above ILP guarantees that for any arbitrary connected graph, the generated set of CHs will be minimal, connected, and provide 100% network coverage. Note that the above formulation can be extended to handle heterogeneous MANETs by making c_{ij} and c_{ji} *independent*. This means that the link between two nodes may exist in one direction only. We use the results obtained from solving the ILP[2] as a baseline against which our proposed algorithm (VGA) for small to medium sized heterogeneous MANETs as well as homogeneous MANETs or even any other related heuristic algorithm, can be compared.

3.2 Control Overhead Analysis of VGA Clustering

In this subsection, we study the communication overhead of VGA and compare it with some related work in the literature. Our objective is to express the control overhead as a function of the number of mobile nodes N, the maximum node degree Δ, and the size of the resulting virtual topology C. Different approximations for finding the minimum connected dominating sets were proposed in the literature. A set of those related approximations are summarized in Table 3.4 as well as our proposed architecture (VGA). In Table 3.4, the metrics of comparison are the cardinality C, which is the size of the generated MCDS, the message overhead, i.e., the number of generated control messages, the time complexity, i.e., the time it takes to create the cluster hierarchy, the message length used in the information exchange in the algorithm, and the number of hops the transmitted information from a node spans before reaching its clusterhead. Note that both [22] and [33] presented two approximation algorithms for finding the MCDS. Also, some of these algorithms study the problem of finding MCDS for special graphs (e.g., the unit disk graph) such as [33] and [31]. For the unit disk graph the problem is known to remain NP-hard; however constant factor approximations are possible in this case.

Now, we show how communication overhead of VGA clustering is evaluated. To analyze the overhead of our clustering algorithm, we divide the algorithm into three parts: (1) Cluster formation and CHs election (OH_1), (2) Cluster update or maintenance due to link addition/deletion (OH_2), and (3) Location management information due to node movements between zones (OH_3). Thus, the total communication overhead of VGA, $OH = \sum_{i=1}^{3} OH_i$. Now, we derive the communication overhead contributed by each factor.

[2] Since this problem is NP-hard, the ILP can only solve small to medium sized problems in a reasonable computation time.

1. Cluster formation and CHs election (OH_1), two rounds of communication are needed to elect a clusterhead (broadcast the eligibility factors, and then the ID of the elected clusterhead node). Since all nodes will be involved in this phase, the cluster formation overhead, OH_1, is $O(N)$.

2. Cluster update or maintenance due to link addition/deletion (OH_2): Effect of mobility can be represented by changes in the underlying VGA graph. For instance, what is the amount of work needed to be done when a link is removed or added or the neighborhood of a node changes. When a link state changes, it will be detected by at most four neighboring CHs. Hence, VGA update overhead due to link addition or deletion is detected by at most four neighboring CHs. In the worst case, four links (neighbors) will either be deleted or added due to a certain zone becoming empty or occupied, respectively. Therefore, the average number of packet transmissions needed for this update is computed as follows: (each CH node directly affected) × (3 neighbors/node) × (2 messages/neighbor) × (1 hop/message) × VGA cardinality (C) = 6 C transmissions. If each clusterhead maintains a global view, then the number of packet transmissions drop to (4 level-2 neighbor nodes directly affected) × (1 messages/neighbor) × (1 hop/message) = 4 packet transmissions, i.e., it takes on the order of $O(1)$.

3. Location management information due to node movements between zones (OH_3): when a node moves to another location, possibly to another zone, its location information needs to be updated. The location management overhead is O(N/Z). Recall that we assumed uniform node distribution. Hence, each zone will have on average (N/Z) nodes.

Table 3.4 Performance comparison for different proposed algorithms. Here *opt* is the size of the optimal MCDS; Δ is the maximum node degree; C is the size of the generated connected dominating set; Z is the number of zones in VGA, V,E: the number of nodes and number of edges in the virtual network graph, respectively.

	Cardinality	Message Overhead	Time Complexity	Msg Length	Hops
[22]-I	$\leq (2ln\Delta + 3)opt$	$O(VC + E + VlogV)$	$O(V + C\Delta)$	$O(\Delta)$	2-hop
[22]-II	$\leq (2ln\Delta + 2)opt$	$O(VC)$	$O(C(\Delta + C)$	$O(\Delta)$	2-hop
[32]	$--$	$O(V\Delta)$	$O(\Delta^2)$	$O(\Delta)$	2-hop
[31]	$\leq 8opt + 1$	$O(VlogV)$	$O(V\Delta)$	$O(1)$	1-hop
[33]-I	$\leq 8opt + 1$	$O(V)$	$O(V\Delta)$	$O(1)$	1-hop
[33]-II	$\leq 8opt + 1$	$O(V)$	$O(V\Delta)$	$O(1)$	1-hop
VGA	$\leq 4.33opt$	$O(V)$	$O(max(\Delta, Z))$	$O(1)$	1-hop

The time needed to finish the VGA clustering scheme is dependent on both the number of zones (Z) and the maximum node degree Δ in each zone. For a large network, the number of zones is comparable to the maximum node degree; hence the time complexity of running VGA is given by $O(max(\Delta, Z))$. This is obvious and can be justified as follows. The set of nodes in each zone elect their clusterhead in two rounds of communication. Each zone clusterhead election process takes $O(\Delta)$ of time. Since a zone needs only to know which other zones are alive, there is a need to propagate a message to the set of all other zones in the network Z. In this case, the time needed to finish the clustering process will be $O(\Delta + Z)$. Therefore, the time complexity of running VGA is $O(max(\Delta, Z))$. Recall that we did not bound the maximum node degree; hence the maximum node degree Δ can exceed the number of zones. Therefore, the time complexity is the maximum of the two parameters Z and Δ. Note that the clustering process needs $O(max(\Delta))$ to finish if global topology information is not needed.

3.3 Analysis of VGA Stability and Connectivity

In this section, we study important properties of the proposed virtual backbone, VGA. Specifically, we analyze connectivity aspects of VGA as well as quantify the effect of node dynamics on the stability of VGA. Since network dynamics evolve with time, we slightly redefine the virtual graph to include the time parameter as follows. Let the communication graph of the virtual topology (VGA) at time t be $G(t)$=$(V(t), E(t))$ where $V(t)$ and $E(t)$ are the set of active CH nodes and the set of active wireless links connecting them at time t, respectively. We assume that G(0) is connected.

3.3.1 VGA Stability Analysis

In this subsection, we focus on the stability of VGA. Recall that in constructing VGA, different parameters like the mobility of the mobile nodes, their remaining energy budget, the number of times a node served as CH, and node's location inside the zone were considered while deciding which nodes can act as CHs. We argue that such a CH selection mechanism improves the stability of VGA. For example, if the nodes with lower mobility are favored for the role of CHs then there will be fewer changes in the set of CHs. Similarly, the fewer the number of times a node served as a CH, or the higher its energy budget, the higher the priority for that node to serve as a CH node. This is done to avoid overloading some nodes and cause them to fail early. Our virtual topology stability is measured by the stability of its constituent members, i.e., the CHs. A CH node stability is measured through two aspects: (a) finding the fraction of time a node remains in the CH state measured over a sufficiently long time window denoted by T (the higher this fraction, the more stable the CH node), and (b) finding the number of times the node served or stopped serving as CH during T (the lower this number, the more stable the network). These two aspects capture the effect of both the number of times a node served as CH and the amount of service time the node acted as a CH during T on the network stability. Let $Q_i(t)$ be the fraction of time a node i assumes the role of CH measured during T, then

$$Q_i(t) = \frac{1}{W}\sum_{m=1}^{W} Y_i(t - m\delta t) \ , \qquad 0 \leq Q_i(t) \leq 1$$

where $W = \frac{T}{\delta t}$ and other involved parameters are defined in Table 4.2. Let $H_i(t)$ be the number of times a node i changes its role as a CH until time t measured over a sufficiently large time window T. Thus,

$$H_i(t) = \frac{1}{2}\sum_{m=1}^{W} |Y_i(t - (m-1)\delta t) - Y_i(t - m\delta t)| \qquad 0 \leq H_i(t) \leq \infty$$

When a node i changes its role as CH more frequently, then its stability decreases. Hence, The stability of the ith CH node with respect to the number of times a node changes its role as a CH node until time t, denoted by $S_i^c(t)$, is defined as:

$$S_i^c(t) = e^{-H_i(t)} \ , \qquad 0 \leq S_i^c(t) \leq 1$$

to indicate that stability decreases with the number of times the CH role changes. Combining the effect of (a) and (b) above can be captured by allowing the ith CH node stability to be represented by:

$$S_i(t) = Q_i(t)e^{-H_i(t)}$$

After the network operates for some time and reaches the time instant t, the stability of the system at time t, $S(t)$, can be calculated as:

$$S(t) = \frac{1}{N}\sum_{i=1}^{N} S_i(t)$$

A higher stability indicates that the nodes change the role of clusterhead at reasonably low frequency, spend more time as a CH, and that, on average, a smaller number of nodes assume the role of clusterhead at a particular instant of time. This means lesser complexity and lower overhead. On the other hand, if the load is evenly distributed among CH nodes, this will help prolong the lifetime of the node and decrease the probability of node failure. Hence, fair load sharing among nodes is an important requirement for stability. We define CH load sharing as the amount of work (time) performed by each CH node. To improve the CH load sharing, those nodes which spend less time being a CH, should be given higher priority in the selection of CHs. The clusterhead load sharing, denoted by $L(t)$, until time t can be measured as:

$$L(t) = \frac{1}{N}\sum_{k=1}^{N}(|Q_k(t) - D(t))|) \qquad 0 \leq L(t) \leq 1$$

where $D(t)$ represents the degree of sharing among nodes to serve as CHs in the network until time t and is found as,

$$D(t) = \frac{1}{N}\sum_{k=1}^{N} Q_k(t)$$

Note that if all nodes fall within one zone, then the value of $D(t)$ will be $1/N$, which means that each node has acted as CH for small amount of time (small share), while if mobile nodes are distributed among all zones, the share of each node serving as CH will be high. Hence, high values of $L(t)$ are normally desired since this indicates that the CH role is being shared among nodes within a zone in a fair manner. In Section 3.4, we show simulation results describing the stability of the proposed virtual topology, VGA, when nodes' dynamics evolve over time.

3.3.2 VGA Connectivity Analysis

A fundamental property of a wireless multihop network is its connectivity [39, 40, 41]. Whereas in cellular systems, it is sufficient that each mobile node has a wireless link to at least one base station for the network to be connected, the situation in decentralized wireless multihop networks is more sophisticated. To achieve a connected network, a wireless multihop path must exist from each node to any other node. Hence, the key element to enhance connectivity is *route redundancy.* The more disjoint paths between two nodes, the better the chance for the network to be connected. Each single node contributes to the connectivity of the entire network. If a node fails or moves, the connectivity might be destroyed. A network is said to be k-Connected if for each node pair there exists at least k mutually disjoint paths connecting them. Two types of disjoint paths are identified: (1) node disjoint, where different paths do not have any node in common; (2) link disjoint, where different paths do not have any link in common. A node disjoint path is also a link disjoint path, but the converse is not true. Hence, we focus on node disjoint paths in this work.

To search for and locate a path(s) over VGA, a CH node needs to calculate the multihop connectivity graph. To do that, the adjacency matrix (one-hop connectivity) of the VGA graph can be manipulated. Note that the adjacency matrix is a method for listing the neighbors of each CH node. The adjacency matrix can be constructed by the exchange of neighborhood information of each CH in a distributed fashion. The adjacency matrix $A = \{a_{ij}\}$ for VGA with a size of $n(t)$ CH nodes, has an entry of 1 at (i,j) to indicate a connection from node i to node j and a 0 entry at (i,j) otherwise. Note that the diagonal elements of A are always equal to zero, i.e. $a_{ii} = 0$, $\forall i$. On the other hand, the entry at (i,j) in the (multihop) connectivity matrix $\mathbb{M} = \{m_{ij}\}$ gives the minimum number of hops needed to connect node i to node j over VGA. Note that $\mathbb{M}^{(1)} = A$ and $\mathbb{M}^{(k)}$ indicates all node pairs that can communicate using k-hop path. Based on the above, a simple algorithm to find the multihop connectivity matrix, $\mathbb{M}$, from the adjacency matrix, A, can be easily formulated as shown in Figure 3.3. The algorithm iterates over all node pairs and for each pair checks to see how many intermediate nodes separate that node pair. At the end of the algorithm execution, $\mathbb{M}$ is the multihop connectivity matrix. Recall that $\mathcal{U}(t)$ represents the number of connected node pairs at time t. Then, the average hop distance, $AHD(t)$, in

```
initialization: Define $\mathbb{M}$ as $n(t) \times n(t)$ matrix, set $\mathbb{M}$=$A$;
for (l=2 to network diameter) do
    for (all node pairs $(i,j)$=(1,1) to $(n(t),n(t))$) do
        if ($i = j$) then
            skip to the next node pair;
        end
        if ($m_{ij} > 0$) then
            skip to the next node pair;
        end
        for (k=1 to $n(t)$) do
            if ($m_{ik} > 0$ and $a_{kj} > 0$) then
                Set $m_{ij}$=$l$;
                exit the loop and go to the next node pair;
            end
        end
    end
end
```

Figure 3.3 Algorithm that finds multihop connectivity matrix $\mathbb{M}$ at a CH node.

the virtual topology of VGA, assuming shortest path routing, can be easily calculated as:

$$AHD(t) = \frac{\sum_{\forall i,j} m_{ij}(t)}{\mathcal{U}(t)}$$

Now we turn our attention to find a measure of how well VGA is connected. We say that VGA is connected if each source-destination (s,d) CH pair is linked by one or more routes. Let p_i be the probability that link i in the graph $G(t)$ is active. Let $\bar{X}(t) = (x_1(t), x_2(t), \ldots, x_e(t))$ be a stochastic process describing the status of the $e(t) = |E(t)|$ links of the graph $G(t)$ at time t such that $x_i(t)$=1 if link i is active at time t and 0 otherwise. The process starts at time 0 with $\bar{X}(0)$. Assuming that links fail independently, then a measure of the connectivity of graph $G(t)$ can be found as:

$$\mathbb{M}_G(t) = \sum_{\forall(s,d)} \sum_{\bar{X}(t)\in\Omega(t)} I^{(s,d)}(\bar{X}(t)) \prod_{k=1}^{e(t)} (p_k)^{x_k(t)}(1-p_k)^{1-x_k(t)}$$

where $\Omega(t)$ is the set of states of the graph $G(t)$ at time t and $I^{(s,d)}(\bar{X}(t))$ is a binary indicator for the existence of a path for each (s,d) CH pair when the system is in state $\bar{X}(t)$ such that it is 1 if a path exists and 0 otherwise. Since we have $\Omega(t) = 2^{e(t)}$ states, calculation of connectivity measure $\mathbb{M}_G(t)$ is infeasible even for small networks. Instead of resorting to approximations, we address the connectivity of VGA in two different aspects. First, we find how many zones will be empty under VGA clustering. Second, we find the probability that a CH node finds itself in *isolation*, i.e., the virtual graph of VGA is disconnected.

Assume that the number of zones in the resulting virtual topology is Z and that the N mobile nodes are uniformly distributed in the Z zones. Under uniform distribution, any node will be equally likely

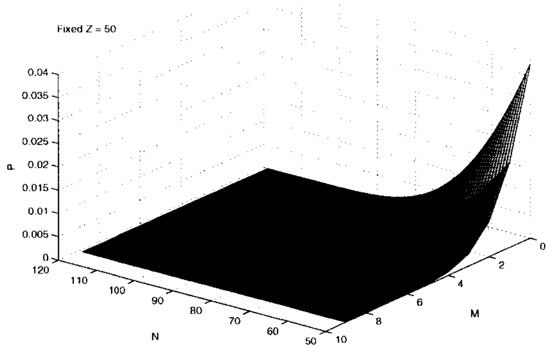


Figure 3.4 The probability of having m empty zones in VGA for Z=50.

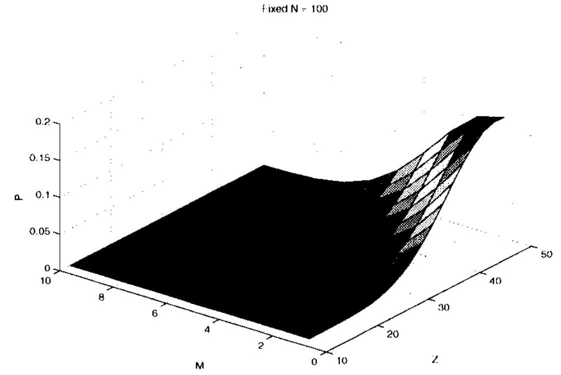


Figure 3.5 The probability of having m empty zones in VGA for N=100.

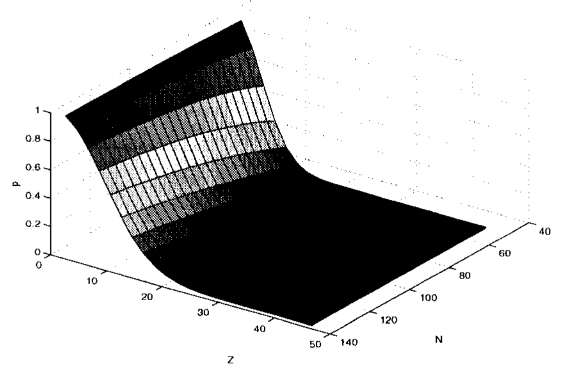


Figure 3.6 The probability of having no empty zones in VGA.

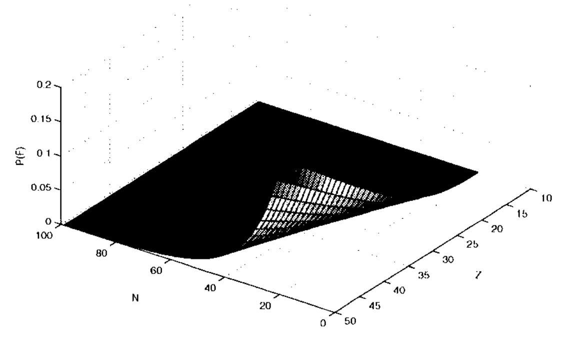


Figure 3.7 Probability of a CH being disconnected in the VGA communication graph.

to fall in any zone with probability $\frac{1}{Z}$. The probability of having exactly m empty zones in VGA, P_m, under the uniform nodal distribution, is easily calculated as:

$$P_m = \frac{\binom{Z}{m}\binom{N-1}{Z-m-1}}{\binom{N+Z-1}{Z-1}}$$

Figure 3.4 shows the value of this probability for various values of N and m when Z=50, while Figure 3.5 shows the value of this probability for various values of Z and m when N=100. In both figures, it can be seen that the probability of having two or more empty zones is very low, e.g., when N=100 and $|Z|$=40 the probability of having 2 empty zones is 0.05320387. Also, when Z=50 and N=60 the probability of having 2 empty zones is 0.00520783. On the other hand, the probability that each zone has at least one node, i.e., none of the zones is empty is given by:

$$P_f = \frac{\binom{N-1}{Z-1}}{\binom{N+Z-1}{Z-1}}$$

Figure 3.6 shows the probability P_f as the number of nodes and number of zones increases. It can be deduced from the figure that as number of nodes increases, the probability of covering the whole area increases. Next, we find the probability that a CH node finds itself in ***isolation***. Recall that in VGA, a CH communicates only with its vertical and horizontal CH neighbors. We will now show that the probability of having VGA disconnected is low. Consider a target zone, and its four neighboring zones as was shown in Figure 3.2(a). Let F be the event that, given that the target zone has at least one node, the four vertical and horizontal zones are empty, i.e., the node is isolated. For the four zones to be empty, no mobile node will fall in these zones under the uniform distribution. Let $P(F)$ be the probability of the occurrence of event F. Then,

$$P(F) = Pr(\text{no node falls in the four zones})$$

$$= Pr(\text{all nodes are in the remaining } |Z| - 5 \text{ zones}) = (\frac{|Z|-5}{|Z|-1})^N$$

Figure 3.7 shows the values of $P(F)$ when N and Z are varied. It is clear that $P(F)$ is very low for large N or $|Z|$, which enhances VGA connectivity.

3.4 Simulation Results

In this section we present the results of experiments corresponding to VGA clustering approach and its performance tradeoffs. The performance methodology adopted here is simulation, where all experiments were performed using using the NS 2.26 simulator [50], except for finding the optimal selection of CHs, which is evaluated using our ILP formulation. We consider a MANET where mobile nodes are initially placed randomly within the fixed-size network area of size 1000×1000 m^2 with the number of mobile nodes ranging between 10 and 300. In homogeneous networks, all mobile nodes have the same transmission range of 150 meters. In heterogeneous MANETs, the long range transmission distance is set to 250 meters, while short range is set to 125 meters so that SR node zone side length is half that of LR node zone. Links have a maximum data rate of 11Mbps. Each ILP problem instance was solved by using the software tool CPLEX [51]. The values of weighting factors in the CH election algorithm were set to $a_i = 0.25$, $i = 1 \ldots 4$. A random distribution is assumed for spatial location of the mobile nodes in the network area. We compare the performance of VGA clustering with the other clustering algorithms proposed in the literature (summarized in Table 3.4). We also compare VGA clustering to the optimal clustering algorithm presented earlier in this chapter. The issues of comparison are: (i) The cardinality of the virtual topology, whose increase results in longer paths, (ii) Control overhead which is the total number of control messages sent per second, (iii) Path optimality, i.e., average path length compared to the optimal path length, and (iv) VGA connectivity.

(i) **CHs cardinality:** As mentioned earlier, the size of the virtual topology needs to be as small as possible. The bigger the size of the virtual topology, the higher the number of message overhead and the

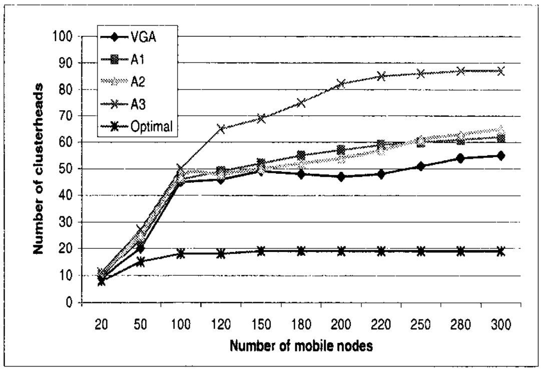


Figure 3.8 CHs cardinality; homogeneous MANETs.

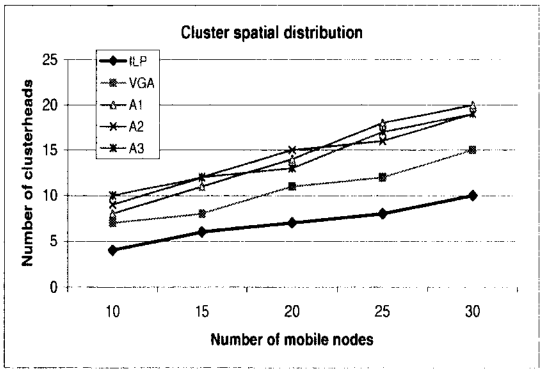


Figure 3.9 CHs cardinality; heterogeneous MANETs.

Table 3.5 0.95 Confidence interval for Figure 3.8 with respect to CHs cardinality.

N	VGA	A_1	A_2
100	45.3±0.32	48.2±0.25	47.4±0.15
150	48.1±0.45	52.2±0.63	53.2±0.53
300	53.3±0.75	62.5±0.12	64.2±0.91

larger the routes. First, we compare the size (cardinality) of the virtual topology computed by different algorithms against VGA. We denote these algorithms by A_1, A_2, A_3, which corresponds to algorithms in [43], [32], and [31], respectively. We randomly distribute a number of mobile nodes ranging from 10 to 300 in the fixed network area such that they must form a connected graph. In the case of optimal zoning, we divided the area into a number of hexagons and found the number of these hexagons, which represents the upper limit on optimal CH cardinality. Figure 3.8 compares VGA to other algorithms in terms of the number of CHs. As Figure 3.8 shows, VGA is able to find a set of CHs that has a smaller cardinality than that obtained by other algorithms. This reaffirms the results of Table 3.4, namely that the cardinality obtained by VGA is closer to the optimal than other algorithms. Table 3.5 shows the confidence interval values for some critical values where the mean and the variance of the mean are reported.

To compare the performance of VGA with the optimal sized MCDS obtained by solving the ILP for small MANETs, we generate connected graphs of 5, 10, 15, 20, 25, and 30 nodes. Each node uses a fixed transmission range of 150m. We used the ILP to solve for MANETs consisting of 5~30 nodes. The ILP was able to find the MCDS for an arbitrary graphs consisting of 10, 15, 20, 25, and 30 nodes in an average computation time equal to 0.32, 13.28, 27.42, 840.63 30, and 1011.5 seconds, respectively. We compare the cardinality of VGA against the optimal number obtained by solving the ILP. Figure 3.9

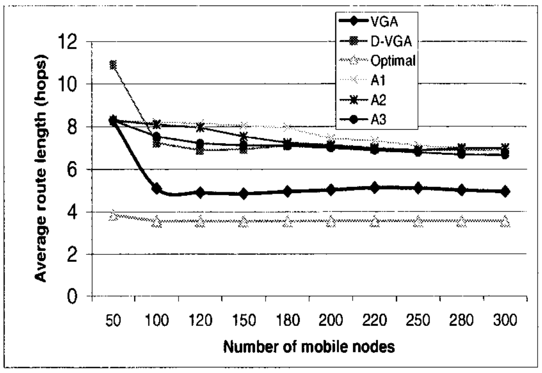


Figure 3.10 Average path length; VGA and D-VGA.

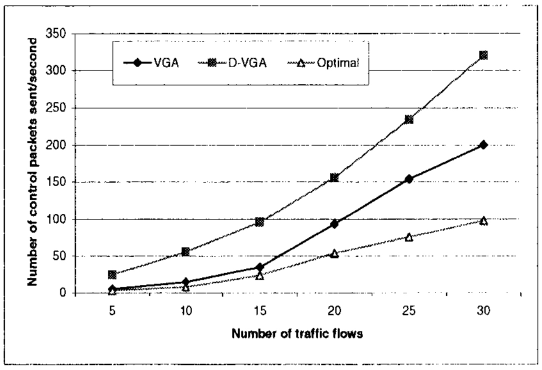


Figure 3.11 Communication Overhead in VGA and D-VGA.

shows this comparison. In almost all cases, VGA is able to obtain cardinalities that are within 50% of the optimal solution. The reason is that VGA uses fixed virtual topology that is made from zones which approximate the optimal hexagon zones. Hence, fewer CHs are needed to cover the network area. Note that the ILP, being computationally expensive, can only solve for networks with a limited number of nodes. Hence the comparison with VGA is also limited to networks with small number of nodes.

(*ii*) **Average route length:** Figure 3.10 shows the average route length of both VGA and D-VGA as well as the optimal average route length when the number of mobile nodes is varied between 10 and 300. The average path length of VGA is closer to the optimal path length than D-VGA. The reason, as mentioned before, is that the cluster (zone) size in VGA is larger than the cluster size in D-VGA. Hence, on the average, fewer zones are encountered between any node pair in VGA routing. These results match the results obtained earlier in the form of closed form expressions. Recall that the transmission range is set to 150 meters. Hence the zone size length in VGA is 67 meter, while in D-VGA it is equal to 53 meters. Hence, the number of zones in VGA is 220, and it is equal to 360 in D-VGA, while it is equal to 60 in optimal zoning. Table 3.6 shows the confidence interval values for some critical values where the mean and the variance of the mean are reported.

Table 3.6 0.95 Confidence interval for Figure 3.10 with respect to average path length.

N	VGA	A_1	A_2
100	5.1±0.12	8.1±0.16	7.9±0.18
150	4.3±0.14	6.5±0.23	6.9±0.13
300	4.2±0.15	6.4±0.12	6.8±0.11

(*iii*) **Control overhead:** It is defined as the total number of control messages sent every second in the network, which are exchanged between CHs. As might be expected, larger cardinalities mean higher communication overhead. Figure 3.11 shows the total number of control messages sent in VGA and D-VGA. As shown in the figure, VGA uses a smaller number of control packets than D-VGA in a constant manner as the amount of traffic in the network increases. This is due to the fact that in VGA, control packets are forwarded in only the possible four directions rather than to the possible eight directions used in D-VGA.

(*iv*) **Connectivity of the CH nodes:** Sometimes, we need all nodes to be connected. For a given network area A and under random node distribution, what are the values of node transmission range (r) and the number of mobile nodes N, that can achieve a certain average network connectivity as it is given by equation (2.1). For the given network area and for the heterogeneous network case with LR nodes, we ran a set of simulation experiments in order to find the number of nodes and the minimum r to obtain a virtual graph with high average network connectivity. As shown in figure 3.12, if we set the transmission radius to 250m when the number of nodes is 100, we obtain a connected graph with average network connectivity that is almost equal to 1. As a result, our initial selection of long range transmission radius of 250 meters is actually based on our simulation results.

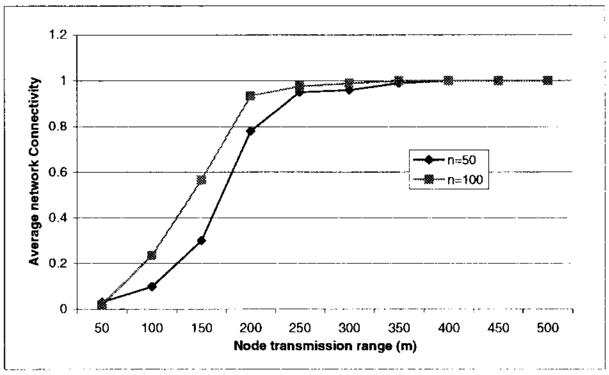


Figure 3.12 Average network connectivity for A=(1000x1000 m^2).

3.5 Chapter Summary

In this chapter, we studied the performance of VGA clustering both analytically and through simulation in both homogeneous and heterogeneous IWNs. For homogeneous networks, with a large number of users, we derived expressions for the number of clusterheads (clusterheads cardinality), VGA worst case path length, and VGA average case path length. We showed mathematically that rectilinear routing can perform better than diagonal routing over VGA. We developed an Integer Linear Program (ILP) that finds the optimal number of connected clusterheads in an arbitrary-connected small- to medium-sized MANETs. We also derived expressions for the communication overhead of VGA clustering. Both analytical and simulation results show that our proposed algorithm pertaining to VGA, though being simple, is close to optimal. We also studied the connectivity and stability features of the proposed virtual topology. The results showed that VGA clustering approach can result in lower control overhead, lower cardinality, and average path length (in terms of number of hops) when compared to other algorithms.

CHAPTER 4 End-to-End Support for Statistical Quality of Service in Heterogeneous Mobile Ad hoc Networks

Future Mobile Ad hoc Networks (MANETs) are expected to support a wide range of real-time multimedia applications. The challenge of QoS provisioning in MANETs is still an open research problem. To provide a complete QoS solution in MANETs, the interaction and cooperation of several components (e.g., QoS routing, QoS scheduling, resource reservation, QoS capable MAC layer, and physical layer) must be studied and quantified. In fact, a new QoS network architecture that suits the nature of MANETs is still needed. The main component in this architecture is QoS routing. Before a connection can be admitted, or any resources can be reserved, a feasible route (a route that has sufficient resources to satisfy the QoS requirements) between a source-destination pair must be found [56]. Packet scheduling is another important component in this architecture. A QoS packet scheduler selects packets to be transmitted from incoming links to outgoing links of a mobile node acting as a router according to the QoS requirements of the sessions sharing that router. In general, QoS guarantees can be either deterministic or statistical. While deterministic guarantees will always be met, and under all circumstances, statistical guarantees allow the guarantee to be met with a certain probability. The characteristics of wireless MANETs preclude any tight bounds on QoS performance measures. Instead, statistical or soft QoS guarantees fit better in this harsh environment [17].

In this chapter, we investigate providing end-to-end statistical QoS guarantees for multimedia applications in MANETs. In particular, we address the important problems of routing and packet scheduling with end-to-end statistical QoS guarantees in heterogeneous MANETs. We present two QoS routing protocols that encompass issues at many layers of the protocol stack, i.e., application, network, MAC, and physical. The two protocols operate on top of the Virtual Grid Architecture (VGA) that was presented in chapter 2.

4.1 Related Work

End-to-end support for Quality of Service (QoS) guarantees is a central and critical issue in the design of future multimedia MANETs. QoS routing is the first step toward achieving end-to-end QoS guarantees in heterogeneous MANETs. However, QoS routing is more difficult in MANETs than in other types of networks for the following reasons. First, the absence of a fixed infrastructure coupled with the ability of nodes to move freely cause frequent route breakage and unpredictable topology changes. Second, the limited bandwidth resource is usually shared among adjacent nodes due to the wireless medium. Third, the nodes themselves can be heterogeneous, thus enabling an assortment of different types of links to be part of the same network. Moreover, QoS routing algorithms require accurate link state (e.g., available bandwidth, packet loss rate, estimated delay etc.) and topology information. Such information are difficult to maintain in this harsh environment.

In fact, QoS routing in MANETs is comparatively a new field, and has just started to receive attention. Designers of QoS routing protocols for MANETs must consider several design issues such as: (1) QoS metric selection (e.g., bandwidth, delay, probability of packet loss, and delay variance (jitter), etc.), power consumption and service coverage area are two other QoS attributes that are more specific to MANETs; (2) QoS path computation methods (e.g., reactive, proactive, etc.) (3) QoS state propagation and maintenance (4) distributed versus centralized algorithmic design (5) the routing architecture (e.g., flat or hierarchical), and (6) scalability for large networks. The QoS routing protocol must also deal with imprecise state information due to node (router) movement and topology changes. Furthermore, a QoS routing scheme for ad hoc networks must balance efficiency and adaptivity, while maintaining low control overhead.

Taking the above difficulties into account, some protocols on QoS routing in MANETs have been proposed. However, when compared to the abundant work on QoS routing for fixed wireline networks, results for QoS routing in MANETs are relatively scarce due to the design challenges inherent in these networks. Since the protocols and parameters employed at the different layers affect QoS, their effects on the QoS performance are hard to quantify. Indeed, most of the proposed routing schemes for MANETs are only QoS aware, but do not really guarantee QoS. Since route selection is based on the desired QoS, the routing protocols were termed QoS-aware. However, the unreliable wireless link and free nodes mobility make it rather impossible to provide strict QoS guarantees in MANETs. Among the QoS routing protocols proposed so far, some use generic QoS measures [62, 63, 67], e.g., they are tuned to a certain performance metric.

In general, QoS routing protocols[1] in MANETs can be classified into four broad categories: (1) Flat-based routing schemes, which are further classified, according to their design philosophy, into five classes: proactive, reactive, predictive, ticket-based probing, and bandwidth-based, (2) Hierarchical-based routing which can be multi-tier network or two-tier network in its simple form, (3) Position-based routing that may or may not use an external location determination server (e.g., Global Positioning System (GPS)), and (4) Power-based routing that optimizes the use of battery limited lifetime. In Flat routing approaches, each node participating in QoS routing plays an equal role. In contrast, hierarchical

[1] A detailed survey of current and future trends in QoS routing in MANETs can be found in [64].

routing usually assigns different roles to mobile nodes that are grouped into multi-level hierarchy. QoS routing that needs help from geographic location system (e.g., GPS) requires each node to be equipped with a GPS card. This requirement is quite realistic today since such devices are becoming inexpensive and can provide reasonable precision. Since energy is also a scarce resource in MANETs, energy or power consumption can be considered as one type of QoS metrics [88]. Power-aware routing in MANETs recently received an extensive research.

Flat-based proactive QoS protocols require each of the nodes in the network to maintain and update routing information [65, 66]. Proactive protocols have many desirable properties, especially for applications with real-time communications. This type of communication requires QoS guarantees, such as low-latency route establishment, alternate QoS path support, and resource state monitoring. Since the satisfaction of such requirement is dependent on the accuracy of the routing information stored in the tables, frequent network topology changes may render this information obsolete and increase the overhead of maintaining state information. Hence, on-demand or reactive QoS routing protocols were proposed [67, 68]. Reactive protocols differ from the conventional proactive routing protocols in that no permanent routing information are maintained at network nodes, thus providing a scalable routing solution in large MANETs. To provide QoS support using one such reactive protocol, AODV [46], a minimal set of QoS extensions has been specified for the RREQ and RREP messages [67]. Another reactive QoS routing protocol that can establish QoS routes with reserved bandwidth in a network employing TDMA is presented in [69]. A shortcoming of the QoS routing protocol in [69] is that it is designed without considering the situation when multiple QoS routes are being setup simultaneously, which may lead to a deadlock situation. On-demand, link state based QoS routing protocols that can also perform admission control were proposed in [70, 71]. The QoS metric is bandwidth. Multiple potential paths are searched at the same time between a source-destination pair to find a path.

Predictive routing detects changes in link status and network topology apriori and uses this information to build stable routes that have low probability of failing [72, 73, 74]. The basic idea of ticket-based probing [75, 76, 77, 78] is to issue a number of tickets based on the available state information in order to search for a low-cost path that satisfies the QoS requirement, i.e., integrating QoS in the flooding-based route discovery. If a QoS routing protocol supports QoS via separate end-to-end bandwidth calculation and allocation mechanisms, it is called bandwidth-based routing. The bandwidth-based routing scheme depends heavily on the use of Time Division Multiple Access (TDMA) medium access scheme in which the wireless channel is time-slotted and the transmission scale is organized as frames (each containing a fixed number of time slots). An end-to-end bandwidth calculation algorithm were proposed in [79, 80]. Using this algorithm, the source node can determine the resource availability for supporting the required QoS to any destination in the ad hoc networks. This approach is particularly useful in call admission control.

The concept of hierarchically structured or multi-clustered routing is utilized to perform QoS routing in MANETs [81, 82]. Very few proposals for cluster-based QoS routing in MANETs have appeared in the literature. An algorithm, called core extraction distributed ad hoc routing algorithm (CEDAR) [81], uses a virtual backbone (core) to perform QoS routing in MANETs. In CEDAR, a set of distributed nodes is dynamically elected to form a virtual core of the network. Each core node maintains the local topology of the nodes in its domain, and also performs route computation on behalf of these nodes. QoS routing in CEDAR is achieved by propagating the bandwidth availability information of stable links in the core graph. The basic idea is that the information about stable high-bandwidth links can be made known to links far away in the network, while information about dynamic links or low bandwidth links should remain local. However, CEDAR works only for small to medium size MANETs, consisting of tens to few hundreds of nodes. Moreover, CEDAR does not maintain multiple paths which is a key requirement for quick route failure restoration and not to mention the effect of route redundancy on the connectivity and stability of MANETs. Using concepts from multi-layer adaptive control, Chen et.al. proposed in [82] an approach for controlling QoS in large ad hoc networks by using multi-clustered organization. The proposed scheme uses bandwidth, represented in terms of time slots, as the QoS metric for nodes grouped in clusters and using a time slotted scheme (TDMA) inside each cluster. A loop free destination sequenced distance vector (DSDV) scheme is used for routing in this architecture.

In the position-based routing, the location of nodes in the network are utilized to improve performance of routing protocols. For example, location information can be used to decrease overhead of routing discovery by limiting the search space for a desired route [14]. The node's location can be provided by some external means such as Global Positioning System (GPS) or the use of beacon devices. Note that there are cases when the GPS does not work (e.g., inside buildings) or the signal can be jammed or blocked maliciously (e.g., battle fields). In this case, a GPS-free approach [15], can be used to allow mobile nodes to approximate their positions using radio strength from a few other nodes. Another example is the Grid location service (GLS) [16], which provides location services by duplicating a node's location in a small subset of other nodes. A GPS-based routing techniques were proposed in [84, 85, 86, 87]. By letting each node maintain a table of the position of all other nodes, a mobile node maintains a snapshot for the network connectivity graph and therefore will be able to compute paths locally without the need for route discovery. An approach to integrate QoS in the flooding-based route discovery process, called positional attribute-based next-hop determination approach (PANDA) [87], discriminates between next-hop nodes based on their location or capabilities. When a route request is broadcast, instead of using a random rebroadcast delay, the receivers seek a delay proportional to their abilities to meet the QoS requirements of the path. The decisions at the receiver side are made on the basis of a predefined set of rules. Thus, the end-to-end path will be able to satisfy the QoS constraints as long as it is intact. A broken path will initiate the QoS-aware route discovery process.

Since MANETs are power-constrained as nodes operate with limited battery energy, *power consumption* is another QoS attribute that is more specific to MANETs. In [88], [89], the authors proposed a routing algorithm based on minimizing the amount of power (or energy per bit) required to deliver a packet from source to destination. The minimum battery cost routing algorithm [90] minimizes the total cost of the route. A Conditional Max-Min battery capacity routing algorithm [90] chooses the route with minimal total transmission power if all nodes in the route have remaining battery capacities higher than a threshold; otherwise routes including nodes with the lowest remaining battery capacities are avoided. Table 4.1 summarizes the different QoS routing paradigms in MANETs. For the sake of comparison, the table also lists our protocols VGAP and VRF, which will be discussed in the next section. VGAP and VRF differ from previous protocols in the sense that both are running over a fixed and stable virtual routing architecture (VGA). Moreover, both protocols capitalize on the cross-layer design approach necessary to improve the performance of MANETs. The comparison between the different routing protocols takes into account many of the design factors mentioned previously like the QoS metrics used, the method used to compute QoS routes, and if the route discovery process returns single or multiple paths.

Table 4.1 Comparison of QoS routing algorithms for MANETs, OD:On-Demand; HYB: Hybrid; C: Clustered; F: Flat; Dist:Distributed; BW:Bandwidth, DLY:Delay, S:Single, M: Multiple, PWR: Power, CA1=TDMA, CA2=CSMA/CDMA.

	Core-based	Ticket-Based	BW-based	Predictive-based	Position-based	Power-based	VGAP	VRF
QoS metric	BW	BW,DLY	BW	BW,DLY	BW	PWR	BW,DLY	PWR,DLY
State maintenance	Local	Local	Global	2-hop info	Local	Local	Local	Local
QoS state propagation	BW changes	Periodic	Periodic	OD	OD	PWR changes	BW changes	OD
Routing class	Dist	Dist	Dist	Dist	Dist	Dist	Dist	Dist
Route computation	OD	OD	OD	OD	OD	OD	HYB	OD
Routing architecture	C	F	F	F	C,F	C	C	C
Single/Multiple paths	S	M	M	S	M	M	M	M
Power issues	No	No	No	No	No	Yes	Yes	Yes
Scheduling issues	No	No	No	No	No	NO	Yes	No
Channel access	CA1	CA1	CA1	CA1	CA1	CA1	CA2	CA2

Most of the previously proposed routing protocols in MANETs address the issue of routing from a single-layer perspective, that is, at the network layer. Recently, a few proposals on inter-layer approaches to QoS provisioning concluded that inter-layer dependencies play a critical role in providing an efficient and comprehensive solution to the QoS routing problem [12], and this view is a key design principle that we capitalize on in the QoS routing protocols proposed in the next section. The use of cross-layer design optimization has been successfully demonstrated in the context of wireless Internet delivery and protocol frameworks for active wireless networks [3]. In fact, layer interdependencies are more pronounced in MANETs.

To illustrate the importance of cross-layer design in MANETs, consider the problem under investigation, namely, QoS routing in MANETs. QoS routing algorithms require accurate link state information (e.g., available bandwidth, packet loss rate, estimated delay, etc.) along with the topology information. The time varying capacity of wireless links and frequent node mobility make maintaining accurate routing information very difficult if not impossible in MANETs. To illustrate, let us understand the kind of interactions that may happen between the three bottom layers in the MANET protocol stack. When a routing protocol computes different routes based on the flow requirements, the set of routes and their flows determine the average load on each link in the network. The selection of a link in a certain route implies also the selection of a certain transmission power (physical) that guarantees transmission quality without affecting the transmission of other neighboring links. A suitable design of a MAC protocol that coordinates among link transmissions and provides a reasonably fair sharing of the wireless link may solve this. However, the MAC may not be able to accommodate the offered load alone as the channel behavior may introduce errors for some transmissions. This means that link quality may degrade and the link metrics exhibit new values, which causes the routing protocol to recompute the routes. This process continues among the bottom three layers for the period of network lifetime. One concludes that MAC and routing protocols depend so strongly on each other, and therefore their joint performance should be optimized, rather than studying the effects of one on the other. Note also that the physical layer parameters have a significant impact on the performance of the MAC layer, and hence on the routing layer. Consequently, the end-to-end network performance is affected by this continuous layers interaction. In fact, the layer interdependencies extend all across the protocol stack.

4.2 QoS Routing in MANETs: A Cross-Layer Approach

In this section, we propose two QoS routing protocols that employ cross-layer design criteria [91] to enhance MANETs performance. In particular, the proposed protocols capitalize on the interdependencies between the physical, MAC, and network layers to provide end-to-end statistical QoS guarantees in heterogeneous MANETs.

4.2.1 VGAP: Quality of Service Routing Protocol for MANETs using VGA

The first protocol, called Virtual Grid Architecture Protocol (VGAP), operates on top of VGA. VGAP employs a routing strategy, which is an extended version of the Open Shortest Path First (OSPF) routing protocol [92] coupled with an extended version of the Weighted Fair Queueing (WFQ) scheduling discipline [93], to provide end-to-end statistical QoS guarantees. The motivation behind using OSPF is that it is currently used in the Internet. Hence, the use of OSPF in MANETs will facilitate the integration between MANETs and the Internet where a MANET can be viewed as one area of the OSPF routing domain. VGAP is able to compute and discover QoS routes over VGA that satisfy end-to-end bandwidth and delay guarantees. The extended version of OSPF, called Mobile OSPF (M-OSPF), is discussed in Section 4.2.1.1, while the extended version of WFQ, called Ad hoc WFQ (AWFQ), is discussed in Section 4.2.1.2. AWFQ scheduling discipline takes into account the time varying characteristics of the wireless channel as will be described in Section 4.2.1.2. The M-OSPF routing protocol interacts with the AWFQ scheduling mechanism to discover and establish end-to-end QoS paths over VGA. As such, the route discovery and resource allocation through scheduling are integrated in the VGAP QoS routing protocol.

Note that some protocols such as CEDAR or ticket-based probing provide on-demand routing, where a route is found based on the pre-known QoS requirements. However, the unpredictable nature of ad hoc networks and the requirement of quick reaction to QoS routing demands make the idea of a proactive protocol more suitable. When a request arrives, the network layer can easily check if the pre-computed optimal route can satisfy such a request. Thus, wastage of network resources when attempting to discover infeasible routes is avoided. However, proactive routing techniques are faced with the challenge of maintaining an accurate state of the network resources. Hence, a *hybrid* approach that makes use of the benefits of both routing classes would be more appropriate in MANETs. This is the routing approach used in VGAP. We now describe the components and operation of VGAP.

4.2.1.1 M-OSPF: An Extension to OSPF in MANETs

OSPF, originally designed for wireline networks [92], has some limitations when employed in all wireless domains like MANETs. First, OSPF has conventionally been designed to work with links that have known bandwidth. This does not carry over to MANETs. MANETs have variations in their available capacities due to the time varying wireless channel characteristics. Second, in OSPF, topology update messages are periodically flooded to the network routers[2]. Periodic flooding consumes more resources in a limited bandwidth environment like a MANET. Reducing flooding frequency without sacrificing accuracy will reduce control overhead and save bandwidth. Hence, updates to the QoS routing tables should be performed only when necessary.

In OSPF, a router (CH) has to learn the identity and resource status of all its neighboring routers. Messages, called Link State Advertisements (LSAs), are used for this purpose. Moreover, since MANETs are typically heterogeneous networks, some mobile nodes may not be QoS capable (i.e., does not have the resources to support *any* QoS guarantees) and they can only support traditional routes. For a node

[2] We will use the terms ClusterHead (CH) and router interchangeably.

to advertise its QoS capabilities, we make use of the T bit, currently unused, in the OSPF options field. A QoS capable node sets this bit in its LSAs. The options field is sent in all OSPF Hello packets and all LSAs. Hence, mobile nodes of different capabilities can be mixed within one OSPF area.

We now present the details of the extended version of OSPF, M-OSPF, designed for MANETs. Analogous to OSPF terms, the MANET geographical region is modeled as one OSPF area since a MANET normally spans a limited region, which is much smaller than an OSPF autonomous system. The set of created zones (see Chapter 2) are modeled as the set of subnets in the OSPF area, and the set of CHs in the zones act as the routers inside the area subnets.

M-OSPF builds its routing tables as follows. First, CHs initialize their routing data structures, i.e., tables or topology database in OSPF terms, by using the OSPF Hello protocol to acquire neighbor CHs. In addition to helping acquire neighbors, hello packets also act as *I'am alive* to let CHs know that other CHs are still functional. Each CH sends an LSA message to provide information to its adjacent CHs (at most four) about initial link states. The set of LSA messages also include information about resources status such as available bandwidth and link processing and queueing delays to neighboring CHs. Once received, the CH stores this information in the QoS routing table and it is used in the QoS route computation algorithm. To address the bandwidth-inefficient periodic message updates of OSPF, M-OSPF employs a threshold based model described later in this section to report any link state changes. By comparing the already established routing tables to reported link states, failed routers (or empty zones) can be detected quickly, and the network topology can be altered appropriately.

M-OSPF uses a *hybrid* criterion in computing end-to-end QoS routes over VGA. When a new request is received by a router (CH), it computes a QoS route locally using the stored QoS routing tables. If the QoS route computation process fails, the router initiates a route discovery process on demand. A user (application layer) sends its request to the network layer for a QoS route with the required metrics in the form $\langle s, d, D_{max}, R_{bw} \rangle$, where R_{bw} is the minimum required bandwidth and D_{max} is the desired maximum end-to-end delay between a source-destination (s, d) pair. From the topological database (routing tables) that was generated earlier from LSAs, each source router calculates a QoS shortest-path tree, with itself as root, such that the tree contains all routes that can satisfy R_{bw} first and then finds the routes that satisfy D_{max}. The QoS shortest-path tree is inspected for an available QoS route. If a QoS route is found, the request is admitted and the route is used. If multiple QoS routes are found, the optimal QoS route (the shortest route that satisfy the QoS requirements), if any, is used and other routes are used as backup routes. If no routes can be computed based on the current state information, an on-demand route discovery process is generated. If the Route Reply (RREP) of the on-demand process returns no QoS routes, the request is rejected. The reason behind using this hybrid QoS routing criteria is to maximize the number of accepted calls (requests).

When a QoS route is broken as a result of a zone becoming empty or the CH fails for any reason, the route maintenance process must be started. A route may also fail if the source node or destination node leaves the original zone and moves to another zone. Note that the former event is rare in dense MANETs. However, mobility of source and destination nodes cannot be controlled. Hence, we need to discover the new location of source or destination nodes after they leave their zones. For this purpose, a simple on-demand query cycle that consists of location request (LREQ)/location reply (LREP) messages can be used, which is handled similar to the RREQ/RREP query messages used by the on-demand ad hoc routing protocols, with the difference being that message broadcast is limited to only at most four neighboring CHs. In addition, query messages are much smaller in size.

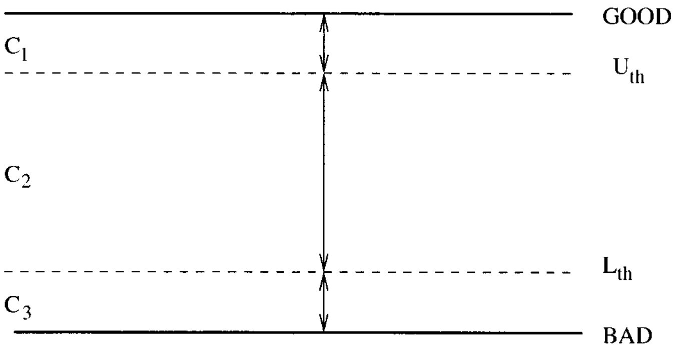


Figure 4.1 A threshold-based scheme for updating link state in M-OSPF.

Since periodic message advertisement in OSPF is a costly process, we now present a simple and efficient scheme that can be incorporated in M-OSPF to limit link state updates and to avoid false and unnecessary updates. Our scheme is a threshold-based triggering scheme, which is characterized by two constant threshold values (Upper threshold (U_{th}) and Lower threshold (L_{th})). Based on the two threshold values, three link classes C_1, C_2, C_3 can be defined (see Figure 4.1). These classes correspond to the current transmission rate (or available bandwidth) on a link. At some particular CH node i, an update is triggered when class boundary is crossed, i.e., if the actual value of the link bandwidth crosses L_{th} or U_{th}. An oscillation around a class boundary, may happen each time the available bandwidth value crosses a class boundary. Hence, the state information in the routing table will be updated more frequently due to the fluctuation around that class boundary. In order to stop such behavior, the scheme is augmented with a ***hysteresis*** mechanism, which requires that change in bandwidth be significant (larger than a certain value) in order to advertise a link state change message. In addition, only a fixed value for the bandwidth (lower bound of the class) is advertised when the link state remains in that class, which may correspond to the IEEE 802.11 data rates of 11 Mbps, 5.5 Mbps, and 2 Mbps.

Note that a link in VGA can fail if a CH becomes congested or fails for other reasons (e.g., runs out of power). Also, when a zone becomes empty or occupied after being empty, a link deletion/addition takes place. When a link is deleted (i.e., link fails) in a certain active route, the route maintenance process in VGAP will be initiated. VGAP can recover from this situation using a route maintenance scheme similar to the one proposed in Section 2.4.3.

4.2.1.2 AWFQ: A Modified WFQ Scheduling Policy for MANETS

In this section, we propose an extension to the Weighted Fair Queueing (WFQ) discipline [94] for MANETs, called Ad hoc WFQ (AWFQ). In VGAP, M-OSPF uses AWFQ fair bandwidth allocations to provide end-to-end statistical bounds on the required delay guarantees. WFQ is an implementation of the Generalized Processor Sharing (GPS) ideal packet scheduler in wired networks [93]. It was shown that WFQ can be lagging behind GPS by no more than $\frac{L_{max}}{R}$, where L_{max} is the maximum packet size, and R is the transmission rate of the link [3]. In WFQ, the concept of Virtual Time $V(t)$ is defined. Each arriving packet is given virtual start and finish times. The packet selected for transmission is the packet with the smallest virtual finish time. In general, if nodes adopt a WFQ-like service discipline and the source traffic is constrained by a leaky bucket, an upper bound on the end-to-end delay guarantees can be provided [94][4]. For ease of reference, Table 4.2 summarizes all notations used in this section. Formally, given a leaky bucket $\langle \sigma, b \rangle$ that constrains the source node traffic where σ denotes the rate at which tokens are accumulated and b is the depth of the token bucket (in bytes), a required bandwidth r on every hop, and a link i propagation delay of τ_i, then the end-to-end delay bound on a route with m hops is given by [94],

$$D_{WFQ} = \frac{b}{r} + \frac{(m-1)L_{max}}{r} + \sum_{i=1}^{m} \frac{L_{max}}{R} + \sum_{i=1}^{m} \tau_i \qquad (4.1)$$

However, when fair queueing algorithms, e.g. WFQ and its variants, are used over wireless networks, the delay bound may not hold [95]. Unlike wireline, wireless links suffer high bit error rates and the channel errors are bursty and location-dependent. In fact, most of the scheduling algorithms that work well in wireline networks do not carry over their desirable properties to wireless environments; they rather lose those properties. Hence, error compensation schemes were devised, essentially for infrastructered wireless packet networks, based on the lead/lag model (see [96] for a useful survey). In lead/lag model, a flow is leading if it is ahead of its error-free reference service, while a lagging flow is a one that lags behind its error-free service due to channel errors.

The proposed AWFQ also uses an error compensation scheme. However, AWFQ differs from all previous schemes in the following ways. First, previous compensation models work for wireless networks with infrastructure where base stations always maintain the scheduler. Base stations use a reliable wireline network to communicate with each other. In MANETs, however, all hops are wireless and therefore packets are prone to bad transmission media on all hops along the path to the destination

[3] Note that WF^2Q [94], another WFQ-variant scheduling discipline, limits the amount of lead of WFQ over GPS. However, our employment of WFQ is generic and can be adapted to WF^2Q.

[4] Reference [94] also provides a detailed description of the operation of WFQ.

Table 4.2 AWFQ Model parameters

Notation	Definition
$p_{(k,i)}$	k^{th} packet of flow i that has a length of $L(p_{(k,i)})$ in bytes.
ψ_i	Service share of flow i
θ	Amount of increment in service share at next node
$W_i^c(t)$	Waiting time counter value of flow i at time t
$C(p_{(k,i)})$	Amount of service time reduction of packet $p_{(k,i)}$ at the next node
C_{max}	The maximum amount of compensation time given to packet $p_{(k,i)}$ at the next node
$A_{(k,i)}, A_{(k,i)}^{(2)}$	Arrival time of k^{th} packet of flow i at nodes 1 and 2, respectively.
$S_{(k,i)}, F_{(k,i)}$	Service start and finish times of k^{th} packet of flow i, respectively.
$S_{(k,i)}^v, F_{(k,i)}^v$	Virtual service start and finish times of k^{th} packet of flow i at node 1, respectively.
$S_{k,i}^{v2}, F_{k,i}^{v2}$	Virtual service start and finish times of k^{th} packet of flow i at node 2, respectively.
V_{new}	New virtual finish time of a donor packet at node 2.
$V(x)$	Virtual time corresponding to real time x
R	Transmission link rate (bps)
$B(t)$	Set of all backlogged (nonempty) flows at time t
$D(t)$	Set of all flows perceiving channel errors and get delayed

node. Second, all previous compensation schemes require that all flows have the same packet size. In our scheme, lead and lag of all flows are defined in terms of virtual time which is flexible enough to deal with packet size differences. Third, the amount of compensation in other schemes is not bounded. If the amount of compensation for lagging flows is large, error-free flows will encounter service degradation that may affect the QoS guarantees of these flows. Although such degradation can be made more graceful in some algorithms [96], it cannot entirely be eliminated. Furthermore, each algorithm has its own specific realization of compensation and thus its performance varies in a wide range in terms of delay bound, throughput guarantee, short term fairness, and gracefulness of degradation. In our extended version of WFQ, we limit the amount of compensation to a maximum value (C_{max}), that is also dependent on the amount of additional delays encountered by the lagging flows.

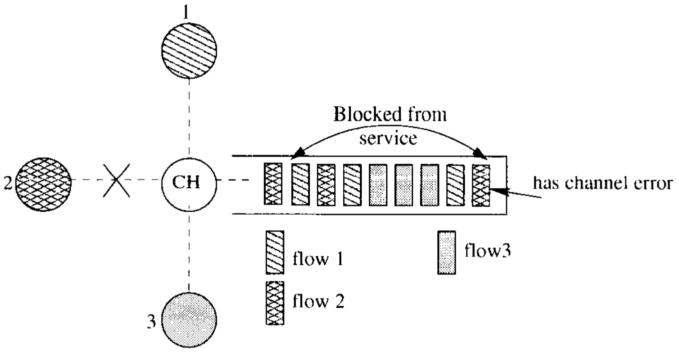


Figure 4.2 AWFQ: A packet experiencing an error can be deferred for a packet with error-free channel.

Under AWFQ, if the Head of Line (HOL) k^{th} packet of flow i, denoted by $p_{(k,i)}$ and with a length of $L(p_{(k,i)})$ in bytes, cannot be transmitted to some other CH node due to channel error (see Figure 4.2 for example), its transmission will be deferred and other flows that have error-free channels will be allowed to transmit. As this packet has a deadline to meet, and because it encountered additional delay at the current node, this additional amount of delay should be compensated for at downstream nodes (CHs) in order for the packet not to miss its deadline. AWFQ does this by modifying (decreasing) the packet virtual finish time at the next node on the path in a manner that increases the probability of meeting the end-to-end delay guarantees. The amount of compensation (or the modified virtual finish time) is computed at the node where the delay occurred, and is carried in the packet header to the next node. We assign each flow i a waiting time counter, $W_i^c(t)$, which stores the amount of additional time delay a flow i packets experience due to channel error at time t. We also define θ as the service share increment of the lagging packet applied at the next node and ψ_i as the service share of flow i.

We now show how the virtual finish time of the packet that incurs additional delay is modified at the next node in accordance with the amount of the additional delay encountered at the previous node. Without loss of generality, we carry the analysis at any two consecutive nodes along the route (current node and next node) and with the aid of the ideal Generalized Processor Sharing scheduler service curve used by WFQ shown in Figure 4.3.

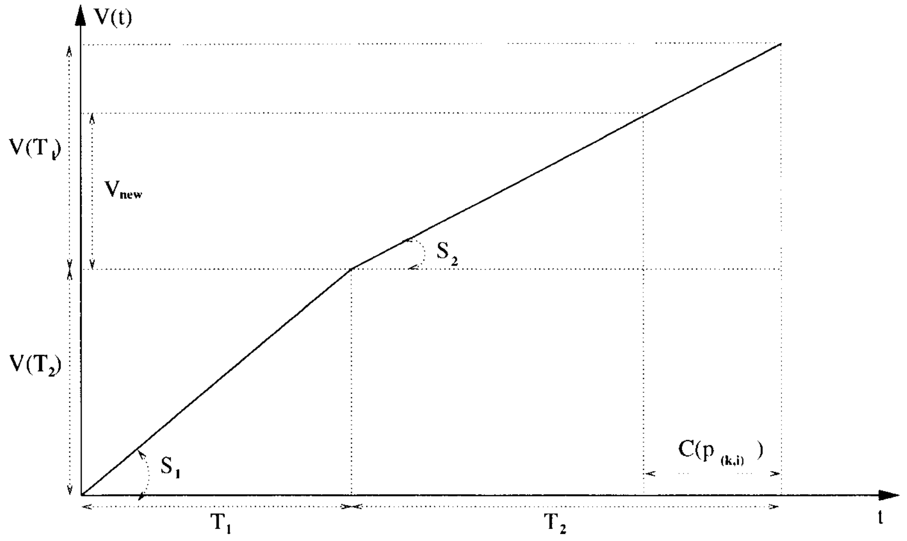


Figure 4.3 GPS service curve at two consecutive nodes in a route

In the service curve, the following parameters are used. We call the current and next node nodes 1 and 2, respectively. T_1 and T_2 are the GPS packet service times at nodes 1 and 2 and are equal to $\frac{L(p_{(k,i)})\sum_{j\in B_1}\psi_j}{\psi_i}$ and $\frac{L(p_{(k,i)})\sum_{j\in B_2}\psi_j}{\psi_i}$, respectively, where B_1 and B_2 are the set of backlogged flows at nodes 1 and 2, respectively. $V(T_1)$ and $V(T_2)$ are the packet virtual service times at nodes 1 and 2, and we assume that $V(T_1)$ is equal to $\frac{L(p_{(k,i)})}{\psi_i}$, S_1 and S_2 are the service curve slopes at nodes 1 and 2 and they are equal to $\frac{1}{\sum_{j\in B_1}\psi_j}$ and $\frac{1}{\sum_{j\in B_2}\psi_j}$, respectively. V_{new} is the new virtual finish time of the packet at node 2. $C(p_{(k,i)})$ is the amount of compensation (reduction in service time) given to packet $p_{(k,i)}$ at the next node and we set $C(p_{(k,i)}) = \min(W_i^c(t), C_{max}, \frac{L(p_{(k,i)})\sum_{j\in B_2}\psi_j}{\psi_i})$, where C_{max} is the maximum amount of compensation given to a packet. Note that $C(p_{(k,i)})$ is upper-bounded by the amount of additional delay encountered by flow i ($W_i^c(t)$) at node 1, and also by the real service time under GPS. We seek to find the new virtual time (V_{new}) of the delayed packet. From Figure 4.3, we have

$$\frac{T_2}{T_2 - C(p_{(k,i)})} = \frac{V(T_2)}{V_{new}}$$

Substituting the values of T_2 and $V(T_2)$ from above, and noticing that ψ_i is the packet service share at the next node, we have

$$V_{new} = \frac{L(p_{(k,i)})}{\psi_i} - \frac{C(p_{(k,i)})}{\sum_{j\in B_2}\psi_j} \tag{4.2}$$

Hence, the new virtual finish time of the delayed packet at the next node is given by

$$F_{k,i}^{v2} = max\{F_{k-1,i}^{v2}, V(A_{k,i}^2)\} + V_{new}$$

Decreasing the virtual finish time of the packet will assign the packet a precedence in scheduling at the new (next) node. The above reduction in service time corresponds to an increment in the service share at the next node (node 2) by a value of θ, which is computed as follows. Since

$$V_{new} = \frac{L(p_{(k,i)})}{\psi_i(1+\theta)}$$

the value of θ can be easily found as:

$$\theta = \frac{1}{\frac{L(p_{(k,i)})\sum_{j\in B_2}\psi_j}{C(p_{(k,i)})\psi_i} - 1}$$

4.2.1.3 End-to-End Statistical Delay Guarantees using AWFQ

In this subsection, we derive the probability that a packet experiencing a channel error, can still meet its end-to-end delay requirement when AWFQ is used. For simplicity, we model the wireless channel using a two state Markov chain where a wireless channel is either in Good(G) or Bad(B) states, respectively. Assume for simplicity and for the sake of analysis that a packet needs one time unit for transmission. Let $1/\lambda_g$ and $1/\lambda_b$ be the average time the channel is in good and bad states, respectively. Then, the transition probability matrix of the Markov chain is:

$$\begin{bmatrix} GG & GB \\ BG & BB \end{bmatrix} = \begin{bmatrix} 1-\lambda_g & \lambda_g \\ \lambda_b & 1-\lambda_b \end{bmatrix}$$

and the steady state probabilities of being in Good/Bad states, π_G and π_B, are given by

$$\pi_G = \frac{\lambda_b}{\lambda_g + \lambda_b} \quad ; \quad \pi_B = \frac{\lambda_g}{\lambda_g + \lambda_b}$$

A packet that has to be transmitted on an erroneous channel continues to wait until its channel state becomes good or the packet misses its deadline and in this case it is dropped. As such, the time elapsed before the successful transmission of an undropped packet follows a geometric distribution at each node with a probability generating function found as follows. At a certain CH node, a packet at the head of the queue will not wait if its channel is in good state (with probability π_G). If it finds the channel in a bad state (with probability π_B), it will wait for a geometrically distributed length of time with probability λ_b. For a path of length m hops and assuming independence between hops, the probability generating function of the total additional waiting time, $G_{D,m}(z)$, is given by:

$$G_{D,m}(z) = (\pi_G + \frac{\pi_B z \lambda_b}{1-(1-\lambda_b)z})^m$$

From $G_{D,m}(z)$, we can find the probability mass function d_n, which is the probability that the total additional delay is equal to n time units and it is found as follows:

$$d_n = \frac{1}{n!}\frac{d^n(G_{D,m}(z))}{dz^n}|_{z=0} \tag{4.3}$$

By finding d_n, we can find all desired performance metrics. To find d_n, let $X = (G_{D,m}(z))^{(1/m)}$ and $Y = G_{D,m}(z)$ and let $q = 1 - \lambda_b$,$p = \pi_B \lambda_b$,$a = p/q$. Then X can be written as:

$$X = \pi_G + (-a + \frac{a}{1-qz}) = (\pi_G + \frac{\pi_B z \lambda_b}{1-(1-\lambda_b)z})$$

The n^{th} derivative of X is easily found as:

$$X^{(n)} = (-1)^n n! \frac{a(-q)^n}{(1-qz)^{n+1}}$$

while the n^{th} derivative of Y is found to be:

$$Y^{(n)}|_{z=0} = \frac{m\sum_{i=1}^{n} X^{(n+1-i)}Y^{(i)} - \sum_{i=2}^{n} Y^{(n+1-i)}X^{(i)}}{X^{(1)}}|_{z=0} \tag{4.4}$$

Substituting equation (4.4) in equation (4.3), we can recursively find the expression for d_n. Using the resulting equation, the probability of having an additional delay time in AWFQ, D_{AWFQ}, which is less than a certain value $\mathbb{W}$ can be found as follows:

$$P(D_{AWFQ} \leq \mathbb{W}) = \sum_{i=0}^{\mathbb{W}} d_i$$

Therefore, the deferred packet can still meet its deadline if ($D_{AWFQ} \leq D_{max} - D_{WFQ} + \mathbb{C}$), where $\mathbb{C}$ is the total amount of applied compensation time for $\mathbb{W}$ and is bounded by C_{max} at each hop, while D_{WFQ} is given by equation (4.1). Hence,

$$Pr(\text{A packet meets } D_{max}) = Pr(D_{AWFQ} \leq D_{max} - D_{WFQ} + \mathbb{C}) \tag{4.5}$$

In Section 4.3, we show that AWFQ is able to meet the deadline for more than 90% of the packets of each admitted flow.

4.2.2 VRF: Quality of Service Routing Protocol for MANETs using VGA

The second QoS routing protocol, called Virtual Routing with Feedback (VRF), addresses load balancing among the routes in the network while providing differentiated QoS guarantees. In practice, some routes may get congested, while others remain un-utilized. This results in poor performance in MANETs. VRF tries to balance the load distribution among various routes while at the same time providing differentiated QoS guarantees. VRF uses on-demand route discovery on top of VGA and has many features similar to that of Dynamic Source Routing (DSR) [42]. However, VRF utilizes a feedback mechanism to discover routes with shorter end-to-end delays or routes with higher energy reserves at individual nodes. In particular, the status of the traffic queues at mobile nodes (called traffic density) and the remaining energy level at these nodes at a certain route are used as indications of the potential delay a packet may incur using that route or the potential lifetime of that route as far as energy is concerned, respectively. VRF selects routes that exhibit shorter end-to-end delays or routes that have higher potential lifetimes among the set of discovered routes over VGA. We are now ready to present the details of VRF.

When a new route request is initiated, the route discovery process of VRF starts. The route discovery of VRF is similar to that of DSR, which uses a Route Request (RREQ)/Route Reply (RREP) query cycle. A CH source node starts the route discovery process to a destination for which it does not already have a route by broadcasting a route request (RREQ) packet across the virtual topology VGA (broadcast is limited to a maximum of four neighbors). A CH node receiving the RREQ may send a route reply (RREP) if it is either the destination or if it has a route to the destination with an updated information about the rest of the route. Otherwise, it forwards the packet to its neighbor CHs. If a route is found, a RREP packet is sent to the source node.

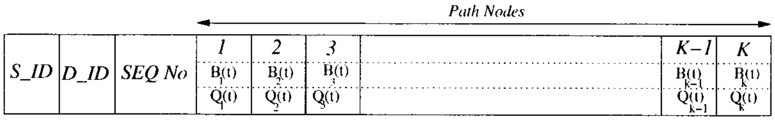


Figure 4.4 Route Reply (RREP) packet format on a path that consists of k nodes. Each node attaches its current queue size and its current remaining battery energy.

As the RREP packet propagates back to the source, each node on the reverse path attaches to this packet its current *traffic density* and its *remaining energy level* along with its ID (see Figure 4.4). The calculation of these parameters is carried out as follows.

- *Traffic density:* Notice that the delay a packet may experience at a certain CH node is dependent, not only on its own queue length, but also on the length of the queues of all its neighbors. This is because the probability that a queue reaching an overflow condition increases when the neighbors are heavily loaded. That is, the probability of overflow is also dependent on the traffic at the neighboring nodes. The traffic queue of a CH node is defined as the average value of the queue length measured over a period of time. For CH node i, it is defined as the average of S samples over a given sample interval,

$$q_i(t) = \frac{\sum_{k=1}^{S} q_i^k(t)}{S}$$

where $q_i^k(t)$ is the kth sample of the queue length and $q_i(t)$ is the average of these S samples at time t. The greater the value of S, the better is the estimation of the traffic at a certain queue. The queue length is averaged over a period of time to eliminate transient effects and to get a more accurate estimate of the traffic at a node over a period of time. This is because the instantaneous queue length may vary rapidly with time due to the bursty and random nature of traffic in MANETs. We define the traffic density of a CH node i as the sum of its traffic queue, $q_i(t)$, plus the traffic queues of all its neighboring CHs. Formally,

$$\mathcal{Q}_i(t) = q_i(t) + \sum_{\forall\, j \in N(i)} q_j(t)$$

where $N(i)$ is the neighborhood of node i, $N(i) \leq 4$, $q_j(t)$ is the size of the traffic queue of node j at time t, and $\mathcal{Q}_i(t)$ is the sum of traffic queues of all the neighbors of node i plus that of node i itself at time t.

VRF protocol requires that each CH node maintains a record of the latest traffic queue estimations of its own traffic queue and of each of its neighboring CHs in a table called the *traffic table*. The traffic table is used to keep the load information of neighboring CHs in VGA. This information is collected by making use of the RREP packet that passes through a CH node. A CH node can also use a message similar to the typical Hello (beacon) message to acquire such information. Hello message contains each node's ID, its current status, and its current neighbors. VRF Hello message can be piggybacked onto the broadcast updates required by VGA. In the former, when a CH node forwards a RREP packet, it attaches to it its identity and its traffic density. Neighbors that receive the RREP packet update the corresponding neighbor's load information in their traffic tables. Finally, we define *Traffic Cost* of a route as the sum of the traffic densities at each of the CH nodes on that particular route. The traffic cost of a route r, between a source-destination (s,d) pair, is given by the following expression:

$$C^{(s,d)}(r) = \sum_{\forall\, i \in r} \mathcal{Q}_i(t)$$

where i is any CH node on route r between (s,d).

- *Remaining energy level:* Collecting the remaining energy level at various nodes follows a similar procedure to that used for collecting the traffic densities. Let $\mathcal{B}_i(t)$ denote the remaining energy level at the battery of CH node i at time t. The *Energy Cost* of a route between (s,d), denoted by $E^{(s,d)}(r)$, can simply be calculated as,

$$E^{(s,d)}(r) = \sum_{\forall\, i \in r} \mathcal{B}_i(t)$$

However, the use of the total remaining energy available on a route as a measure of which route has the maximum remaining energy available is not practical, since it does not reflect directly on the lifetime of each mobile node along that route. Thus, we would like to route packets along the path with the maximum of the minimal fraction of remaining energy, called *Max-Min route*. A Max-Min route is the route where individual nodes in the route have relatively higher energy than other routes. When deciding which route to use, the source node will attempt to avoid the route with individual nodes having the least energy capacity among all nodes in all possible routes. This smoothes the use of the battery of each mobile node. This will also minimize the variance in node power levels.

In another variation to VRF, we simply replace the traffic density at each intermediate CH with the average queue size of that CH node. Hence, the traffic cost of a route will be simply the sum of the average queue sizes (the total number of queued packets) at the CHs constituting the route, i.e.,

$$C^{(s,d)}(r) = \sum_{\forall\, i \in r} q_i(t)$$

We believe that this simple variation will decrease the communication overhead of VRF and make it much simpler. However, this simplicity comes at the expense of lower performance of VRF as will be shown in the simulation results. We refer to this variant of VRF as VRF-Q.

In both VRF and VRF-Q, the route discovery process may return multiple routes. Let $\mathbb{R}$ be the set of these routes between any (s,d) pair. The information returned on each route r, $r \in \mathbb{R}$ that are embedded in the RREP packet is used by the source node to determine which route will be used in order to deliver its packets. For the least delay path, the decision is based on the number of hops and the traffic cost of that route. Whenever a choice has to be made, we select the route with the minimum traffic cost. This route is the one that results in the least delay as it is the least congested route. In case of a tie between two routes that exhibit the same traffic cost, the route with the least number of hops is used. If the QoS metric required is the power consumption, the route with higher energy capacities of individual nodes is selected for long lasting sessions. Hence, a source node selects as the primary route, the route with the least $C^{(s,d)}(r)$ for delay-constrained flows, and the route that satisfies the Max-Min energy criterion for long lasting flows. All other routes are used as backup routes for both metrics. Hence, the primary route for the delay metric, $\mathcal{D}$, and for the energy metric, $\mathcal{E}$, are selected as:

$$\mathcal{D} = \arg\min_{r \in \mathbb{R}} (C^{(s,d)}(r))$$

$$\mathcal{E} = \arg\max_{r \in \mathbb{R}}\{\min_{i \in r}(\mathcal{B}_i(t))\}$$

Notice that the route $\mathcal{E}$ represents only the route that has high probability of survivability (route with maximum lifetime). When the primary route fails, a backup route can be used immediately. Hence, the reliability of the network is enhanced.

4.3 Performance Evaluation

The VGAP and VRF protocols were simulated using the NS 2.26 simulator [50]. A heterogenous MANET in an area of size (2000m×2000m) with a variable number of mobile nodes was simulated. The long range transmission distance is set to 250 meters, hence zone side length is $x_l = 112meter$. Since a subzone side length is $x_s = x_l/2$, the value of short range (r_s) is set to 125 meter. Mobile nodes were initially placed randomly within the fixed-size network area. Mobile nodes roam around according to a certain mobility model. We study the effect of four mobility models on the network performance in this section. The number of mobile nodes is set to a default value of 200 unless otherwise stated. Links have a maximum data rate of 11Mbps. Calls arrive to the system according to a Poisson process with an arrival rate that was varied to control the traffic load into the network. The call holding time is exponentially distributed with a mean of 60 seconds. Unless otherwise specified, the default offered traffic load was set to 0.8. Each call is specified as a randomly chosen source-destination (*s-d*) pair. The packet sizes are exponentially distributed with a mean of 512 bytes. Our energy consumption model is based on commercial settings. For every transmission and reception of packets, the energy level is decremented by a specified value, which represents the energy usage for transmitting and receiving. Power requirement for receiving, transmitting, or idle modes follows the energy model presented in chapter 2. Since it is impossible to evaluate the behavior of the network if the nodes generating traffic (traffic nodes) run out of energy before the end of simulations, we give traffic nodes infinite energy. In another setting, we limited the initial node energy to study the effect of energy savings on network lifetime for various protocols. We model GPS as consuming 0.0033W, the amount of power necessary for reporting location every 10 seconds, since VGAP does not require constant position information. In order to quantify energy consumption, we define the mean power consumption per node, Me, as follows. At the start of the simulation the nodes have a total initial energy of E_0. After time t, the remaining total energy of the n nodes is E_t, which is calculated as $Me = \frac{E_0 - E_t}{n*t}$. The value of Me will be used to measure the number of alive nodes after some time t under various protocols.

Two types of calls were generated, audio and video. The requested bandwidth of each call is uniformly distributed between 16 and 64 Kbps for audio and 0.5 to 2 Mbps for video. Audio calls constitute 40% of the total generated calls and the rest are video calls. In the two state Markov chain wireless channel prediction model, we set $\lambda_g = 0.1$ and $\lambda_b = 0.9$ per average packet transmission time.

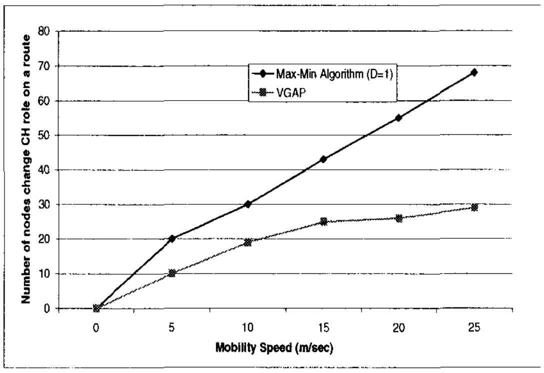


Figure 4.5 Percentage of CH nodes that change their role with mobility speed, r_l=250m and N=100.

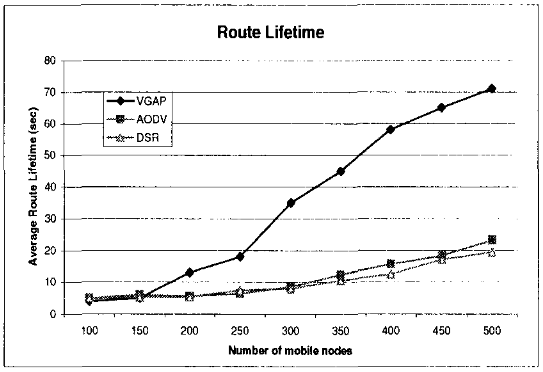


Figure 4.6 Route lifetime vs the number of mobile nodes for A=(2000x2000 m^2, and mobile speed 15m/sec.

For the link state update model, we set $((L_{th})$=0.3 and $(U_{th}))$=0.7 of the link rate. In another setting, we also vary those two values to study the behavior of the model and its effect on the performance of VGAP. We average each result over 30 simulation runs. We show the results of our simulation model in terms of the following metrics:

Route Stability/Lifetime: In chapter 3, we demonstrated the network stability under VGA clustering. In this chapter, we evaluate route stability in VGAP by measuring how many times, on average, a CH node on a route changes its role taken over all routes used by the protocol. Recall that there is an overhead associated with the CH role change. In VGA, as long as the zones are not empty, the route will remain stable. For this part, we fixed N= 100 and we varied the mobility speed of the PMM mobility model (Chapter 2) and count the number of CH role changes. In Figure 4.5, we compare our algorithm to another clustering algorithm (Max-Min D-Cluster Formation with D=1) proposed in [30]. Max-Min D-Cluster algorithm tries to re-elect the same CHs when possible to enhance cluster stability. As shown in the figure, the stability of VGA allows VGAP, on average, to maintain a lower percentage of nodes changing the CH role. We also measured the ***Route Lifetime*** in VGAP. Figure 4.6 shows the average lifetime of a route in VGAP compared to AODV [46] and DSR [42] protocols that are already implemented in ns-2. We vary the number of mobile nodes between 100 and 500 to observe the effect on the route lifetime. Since the physical area remains the same, a larger number of mobile nodes means a higher node density. VGAP is able to achieve the highest route lifetimes among all protocols. This shows that VGA stability reduced the possibility of frequent route breakage. We also measured the average lifetime extension of the VRF protocol when compared to DSR protocol. Figure 4.7 shows that VRF is able to allocate paths that exhibit higher lifetimes than DSR due to the stability of the underlying VGA routing architecture.

Packet Delivery Ratio: We define the packet delivery ratio as the ratio between the number of

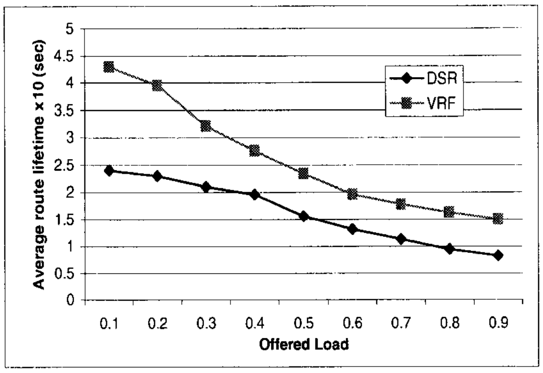


Figure 4.7 Average route lifetime in the VRF protocol.

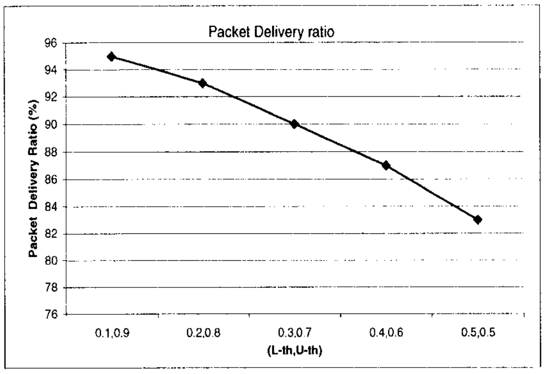


Figure 4.8 Packet Delivery ratio versus the link state update thresholds.

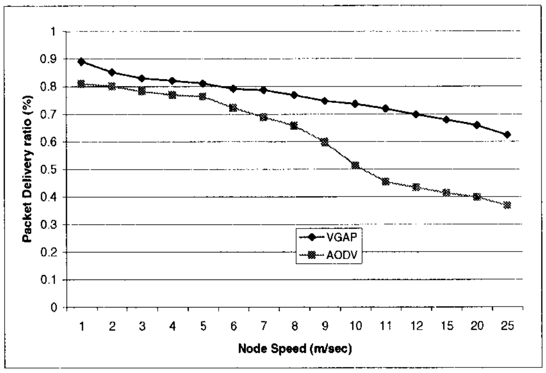


Figure 4.9 Packet delivery ratio as an indication of network reliability.

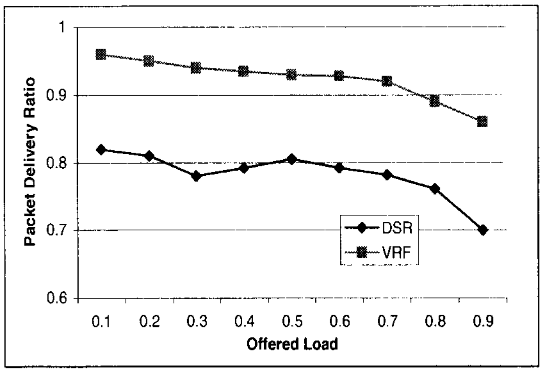


Figure 4.10 Packet delivery ratio in the VRF protocol.

packets received by the destination and the number of packets generated by the application layer of source nodes of admitted calls. Packet delivery ratio is important as it describes the loss rate that will be seen by the transport protocols, which in turn affects the maximum throughput that the network can support. It is also an indication of the network reliability and the ability of the network to support more QoS routing calls. Figure 4.8 shows the packet delivery ratio using M-OSPF in VGAP and employing the proposed threshold-based link state advertisement model. The two thresholds (L_{th},U_{th}) are changed from (0.1,0.9) to (0.5,0.5), respectively. VGAP is able to achieve high packet delivery ratio and maintain acceptable levels when the link updates increase, i.e., increase in control traffic. This indicates that VGAP is able to use primary and backup routes efficiently to deliver more packets, where backup routes are instantly used when the primary route fails. We then varied the node speed under the PMM mobility model (see Chapter 2) and measured how many packets are successfully received at the destination compared to AODV protocol. As shown in Figure 4.9, VGAP is able to maintain acceptable levels of packet delivery ratio even with higher node mobilities than AODV. This

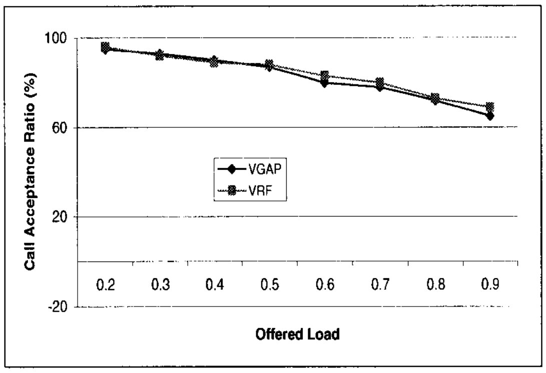


Figure 4.11 Call acceptance ratio in the VGAP routing protocol as offered load increases.

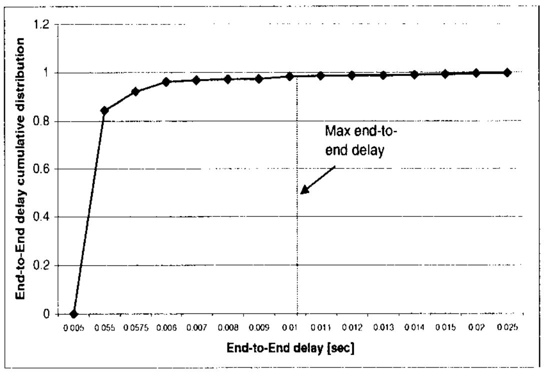


Figure 4.12 Cumulative distribution of end-to-end packet delay in VGAP.

is due to the extended lifetime of the discovered routes, thanks to the VGA stability. In this figure, when v=3 m/sec, the 0.95 confidence interval of VGAP is 0.85±0.072 while it is 0.78±0.081 for AODV. To further consolidate the impact of VGA stability on packet delivery ratio, we measured the packet delivery ratio in VRF and DSR for various offered loads in Figure 4.10. VRF, running on top of VGA, is able to deliver more packets to destinations than DSR even for higher offered loads. One concludes from these results that the stability of the underlying routing architecture enabled VGAP and VRF to obtain better performances than related protocols.

Call Acceptance Ratio: Call acceptance ratio is defined as the number of successfully accepted route requests divided by the total number of requests generated in the network for the length of the simulation period. Note that for each admitted session, the routing protocol was able to allocate a route with the required QoS guarantees. Figure 4.11 shows the call acceptance ratio versus the network offered load in VGAP and VRF. Since VGAP finds routes that exhibit long lifetimes due to its underlying stable routing architecture, it is expected that the call acceptance ratio will be high. Figure 4.11 shows that as the offered load increases, the call acceptance rate drops with a reasonable rate. For higher levels of offered loads, both VGAP and VRF still accept more than 65% of the calls. In this figure, when offered load= 0.4, the 0.95 confidence interval of VGAP is 0.92±0.0032 while it is 0.91±0.0081 for VRF.

End-to-end Delay: The end-to-end packet delay is also studied in both VGAP and VRF protocols. When a link is unreliable, the node fails to forward packets, causing packet drops or longer delays. At low traffic load, nodes rarely experience congestion but often experience broken links. Therefore, packets that need to be re-routed will be queued and therefore encounter longer delays. The packet end-to-end delay was set to 10 msec. VGAP was able to satisfy the delay requirements for most of these packets (more than 90% of the packets) as shown in Figure 4.12. The figure shows the cumulative probability distribution function of the end-to-end delay of the generated packets in the network. We

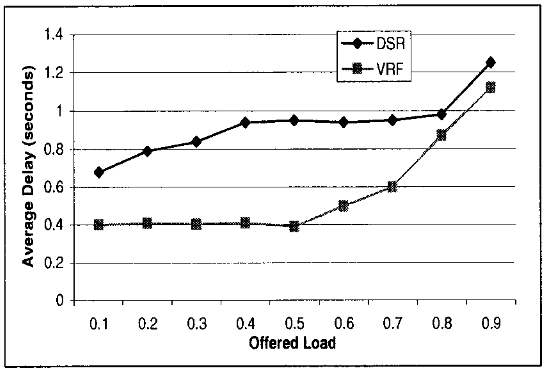


Figure 4.13 Average delay using the VRF protocol versus the DSR protocol.

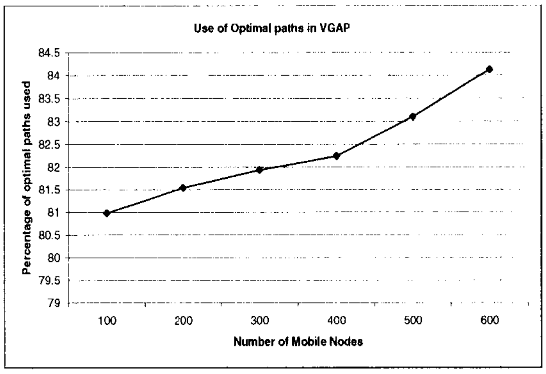


Figure 4.14 Percentage of optimal paths used in the VGAP protocol.

notice that most packets were delivered successfully within their required delay bounds. On the other hand, Figure 4.13 shows the the average delay of VRF compared to DSR. We observe that VRF is able to find routes that incur shorter delays than DSR. VRF criterion of selecting least delay routes enables it to always select routes with less end-to-end delays than DSR. In this figure, when offered load=0.9, the 0.95 confidence interval of VRF is 1.17±0.021 while it is 1.26±0.048 for DSR.

Path Optimality: The path with the minimum number of hops which still satisfies the required QoS metrics is an optimal path since it uses less resources. Recall that an optimal path is the shortest path that satisfies the QoS guarantees for each request. After determining the neighborhood, we calculate the shortest path routing using the well-known Dijkstra algorithm. The number of optimal paths used by VGAP as a percentage of the set of all optimal paths available for routing is shown in Figure 4.14. Note that VGAP, which employs M-OSPF and AWFQ together, is able to use more than 80% of the optimal paths while serving various QoS requests. This efficient use of the resources also explains why VGAP was able to achieve high packet delivery ratio.

Communication Overhead: In chapter 2, we have seen the profound effect of VGA clustering on reducing the communication overhead in terms of the total number of control packets. We now show that this also results in a significant reduction of routing control packets overhead in VGAP. Figure 4.15 shows the generated control overhead compared to a variable size clustering (adaptive clustering) algorithm proposed in [24]. We varied the number of mobile nodes from 100 to 500 and measure the control traffic generated in VGAP and the one in [24]. A substantial saving in control overhead is achieved in VGAP due to the use of the fixed size clustering scheme. This is due to the reduction of CHs cardinality and CH stability as was demonstrated in Chapter 2. As the node density in the network increases, VGA tends to be more stable and the discovered routes tend to last longer, which lowers the probability of route breakage and the extra overhead associated with it. Recall that control messages are generated

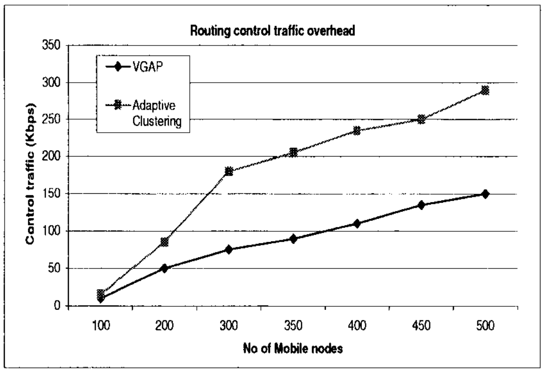


Figure 4.15 Control overhead of VGAP versus adaptive clustering that was proposed in [24].

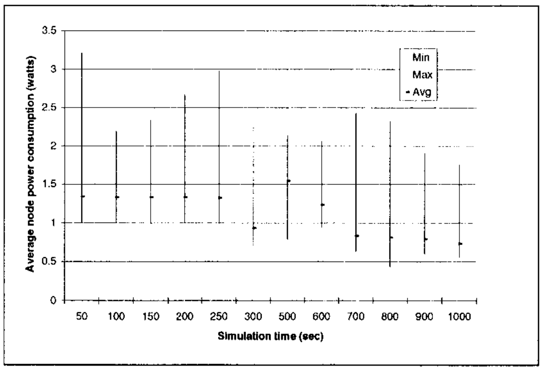


Figure 4.16 Average node power consumption in VGAP.

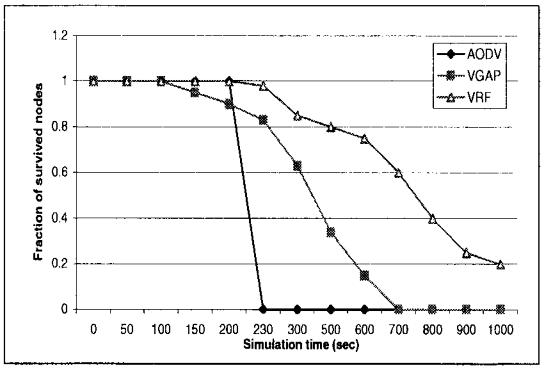


Figure 4.17 Comparison of non-zero energy node fraction over time: VGAP, VRF, and AODV.

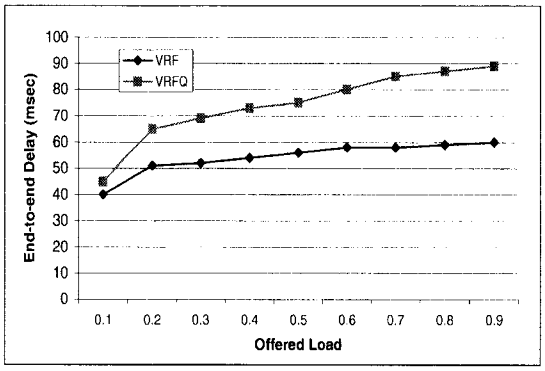


Figure 4.18 Comparison of VRF and its variant VRF-Q in terms of end-to-end delay.

by VGAP when a change to the underlying VGA occurs, i.e., the route breaks or route maintenance is required. Coupled with low control overhead is scalability, which is an important factor that refers to the adaptability of the protocol to larger networks. Figure 4.15 shows that VGAP scales almost linearly in terms of the control overhead as the node density in the network increases.

Energy Consumption: We now examine how energy savings of VGAP can extend the network operational lifetime. In this experiment, we set E_0 of each node to 8000J. We also fixed the number of nodes n to 350. The mobility model used was PMM. Due to their use of the stable and energy-efficient underlying VGA, it is expected that both VGAP and VRF will result in extending the network operational lifetime. Hence, we ran a set of experiments and in each experiment, we measured the mean power consumption per node, Me, at different time instances of the simulation time. We also measured the minimum and maximum power consumption at that time instance among all nodes. Figure 4.16 shows the average node consumption in VGAP along with the minimum and maximum node power consumption at a set of time instances averaged over a set of 30 experiments. It is noted from the figure that VGAP managed to maintain low average power consumption per node. This observation indicates that VGAP is able to extend the network operational lifetime more than other protocols. To validate this observation, we also measured the fraction of survived nodes at each time instance. As shown in figure 4.17, VGAP is able to save 50-60% more energy than AODV, while VRF can save about 60-70% more energy than plain AODV. These values were obtained by finding the total power consumption used by each protocol. The figure also shows the fraction of the network with remaining energy over time when nodes move at speed of 12m/s. All nodes running AODV run out of energy at the same time, around 230 seconds. We can also see that VRF balances energy use more evenly among nodes than AODV and VGAP. For example, at time around 500 seconds at least 80% of VRF nodes are still alive while at most 40% of VGAP nodes are alive and 0% nodes alive in AODV.

Performance of VRF and VRF-Q: Finally, we compare the performance of VRF and VRF-Q in terms of end-to-end delay. We varied the offered load and measured the end-to-end delay. We fixed N to 200 nodes. Our objective is to show that using traffic density as a measure of congestion at a node is better than merely using the average queue size at the node. As shown in Figure 4.18, as the offered load increases, the VRF routing protocol is able to find routes with shorter end-to-end delays than VRF-Q does. The reason is that as the load increases, the node's queue length in VRF-Q gets longer and hence higher delays are incurred in VRF-Q. On the other hand, VRF is able to locate routes that avoid nodes prone to future high traffic by catering for neighborhood conditions, and hence selects routes that tend to develop shorter queues than VRF-Q. This results in shortening the end-to-end delay in the system.

4.4 Chapter Summary

In this chapter, we developed two QoS routing protocols for heterogeneous MANETs that use VGA of Chapter 2 as their routing structure. The first protocol, called Virtual Grid Architecture Protocol (VGAP), uses an extended version of the OSPF routing protocol, called Mobile OSPF (M-OSPF) and an extended version of WFQ scheduling policy, called Ad hoc WFQ (AWFQ) to discover Qos routes with statistical end-to-end delay and bandwidth guarantees. Besides being fully distributed, VGAP requires little communication overhead. VGAP capitalizes on the interplay among layers to enhance the network performance. The second QoS routing protocol, called Virtual Routing with Feedback (VRF), utilizes a feedback mechanisms to discover routes with smaller end-to-end delays or routes with higher energy reserves. Our results show that both VGAP and VRF are able to support QoS in MANETs by allocating QoS routes over the stable VGA, which results in high packet delivery and call acceptance ratios. Moreover, the operational network lifetime is increased under VRF due to the load balancing mechanism used in the protocol.

CHAPTER 5 Data Aggregation and Routing in Wireless Sensor Networks: Optimal and Heuristic Approaches

Wireless Sensor Networks (WSNs) are mainly characterized by their limited and non-replenishable energy supply. Thus, a fundamental challenge in the design of WSNs is to maximize their lifetimes. To extend the network lifetime, power management and energy-efficient communication techniques are necessary [10].

In this chapter, we consider the problem of correlated data gathering in WSNs. Our objective is to maximize the network lifetime by utilizing data aggregation and in-network processing techniques. We particularly focus on the *joint* problem of optimal data routing with data aggregation enroute such that the above mentioned objective is achieved. We present Grid-based Routing and Aggregator Selection Protocols (GRASP), a scheme for WSNs that can achieve low energy dissipation and low latency without sacrificing quality. GRASP embodies optimal (exact) as well as heuristic approaches to find the minimum number of aggregation points while routing data to the BS such that the network lifetime is maximized.

5.1 Related Work

Due to the correlation present among different sensors' readings in WSNs, it is expected that communication approaches that take into account this correlation, e.g. data aggregation and in-network processing, will outperform traditional approaches. The main idea of data aggregation and in-network processing approaches is to combine the data arriving from different sources (sensor nodes) at certain aggregation points (or simply aggregators) enroute, eliminate redundancy, and minimize the number of transmissions before forwarding data to an external base-station (BS). Given that, in WSNs, the communication cost in terms of energy consumption is several orders of magnitude higher than the computation cost [127], in-network data aggregation can achieve significant energy savings. In fact, aggregation scheme comparison studies [97, 98, 102, 122] conclude that enhanced network throughput and more potential energy savings are possible using data aggregation and in-network processing in WSNs.

Aside from the task of efficient design of data aggregation algorithms, the task of finding and maintaining routes in WSNs is nontrivial. Many routing, power management, and data dissemination protocols have been proposed for WSNs. In general, routing in WSNs can be broadly divided into *flat-based* routing, *hierarchical-based* routing, and *adaptive-based* routing depending on the network structure. Furthermore, these protocols can be classified into multipath-based, query-based, negotiation-based, or location-based routing techniques depending on the protocol operation (a detailed survey on routing techniques in WSNs can be found in [99]).

Under the flat-based routing category, routing decision in Sequential Assignment Routing (SAR) [100] is dependent on three factors: energy resources, QoS on each path, and the priority level of each packet. The objective of SAR algorithm is to minimize the average weighted QoS metric throughout the lifetime of the network. In [97], C. Intanagonwiwat et. al. proposed a popular data aggregation paradigm for WSNs, called directed diffusion. Directed Diffusion is a data-centric (DC) and application-aware paradigm in the sense that all data generated by sensor nodes is named by attribute-value pairs. The base station requests data by broadcasting interests. Interest, describing a task required to be done by the network, diffuses through the network hop-by-hop and drawing gradients that satisfy the query (interest) towards the BS. When interests fit gradients, paths of information flow are formed from multiple paths and then some paths are reinforced so as to prevent further flooding according to a local rule. In order to reduce communication costs, data is aggregated on the way. The goal of directed diffusion is to find a good aggregation tree which gets the data from source nodes to the BS. The *Minimum Cost Forwarding* Algorithm (MCFA) [101] exploits the fact that the direction of routing is always known, that is, towards the fixed external base-station. Hence, a sensor node need not have a unique ID nor maintain a routing table. Instead, each node maintains the least cost estimate from itself to the BS. The focus of [101] is on routing, but no data aggregation was performed.

In hierarchical or cluster-based routing, Heinzelman, et. al. [102] introduced a hierarchical clustering algorithm for sensor networks, called Low Energy Adaptive Clustering Hierarchy (LEACH). LEACH is a cluster-based protocol, which includes distributed cluster formation in which the sensors elect themselves as clusterheads with some probability. The algorithm is run periodically, and the probability of becoming a clusterhead for each period is chosen to ensure that every node becomes a clusterhead at least once within 1/P rounds, where P is the desired percentage of clusterheads. The remaining sensors join the cluster of the clusterhead that requires minimum communication energy. In LEACH, CH nodes compress data arriving from nodes that belong to the respective cluster, and send an aggregated packet to the BS. While data processing can take place anywhere in the network, LEACH can achieve energy savings by processing data at its cluster head nodes. In [107], results in stochastic geometry were used to derive solutions for the values of two parameters (number of hops to reach CH and probability of becoming a clusterhead) of a modified max-min d-clustering algorithm in order to minimize the total energy consumption in the hierarchical network structure. In [103], an enhancement over LEACH, called Power-Efficient Gathering in Sensor Information Systems (PEGASIS), was proposed. PEGASIS is a near optimal chain-based protocol where nodes only communicate with their closest neighbors and they take turns in communicating with the BS. Two hierarchical routing protocols called TEEN (Threshold-sensitive Energy Efficient sensor Network protocol), and APTEEN (Adaptive Periodic Threshold-sensitive Energy Efficient sensor Network protocol) are proposed in [104] and [105], respectively. These protocols were proposed for time-critical applications and they use a set of threshold values to control the trade-off between energy efficiency and data accuracy. In [106, 108, 110] protocols were introduced to compute an energy-efficient subnetwork, namely the minimum energy communication network (MECN) for a certain sensor network. In Geography Informed Routing (GIF) protocols [109], the network area is first divided into fixed zones, and nodes collaborate with each other to play different roles in order to save energy. However, neither routing nor data aggregation were discussed in [109]. A Two-Tier Data Dissemination (TTDD) approach [111] provides data delivery to multiple *mobile* bas-stations. In TTDD, each data source proactively builds a grid structure which is used to disseminate data to the mobile sinks by assuming that sensor nodes are stationary and location-aware.

Under the adaptive-based routing paradigm, Heinzelman et.al. proposed a family of adaptive protocols called Sensor Protocols for Information via Negotiation (SPIN) [112, 113]. Nodes running SPIN assign a high-level name, called meta-data, to completely describe their collected data, and then perform meta-data negotiations before any data is transmitted to prevent the transmission of redundant data. Table 5.1 outlines the major differences between flat-based, hierarchical-based, and adaptive-based routing approaches in WSNs using different metrics. The table also lists the advantages and disadvantages of each routing paradigm.

Table 5.2 compares SPIN, LEACH, and the Directed Diffusion routing techniques according to different parameters. It is noted from the table that Directed Diffusion shows a better approach for energy-efficient routing in WSNs due to the use of data aggregation and in-network processing techniques.

The fault tolerance (resilience) of a routing protocol can be increased by using and maintaining multiple paths between the source and the destination at the expense of an increased energy consumption and traffic generation [114] [115] [116]. In query based routing, a destination node, e.g., the BS, propagates a query for data through the network. Nodes having a matching data to the query will send it back to the requesting node [97] [117]. In Negotiation based routing protocols, a high level data descriptors are used in order to eliminate redundant data transmissions through negotiation [112]. A position based algorithm called SPAN [118] selects some nodes as coordinators based on their positions. The coordinators form a network backbone that is used to forward messages. A node should become a coordinator if two neighbors of a non-coordinator node cannot reach each other directly or via one or two coordinators (3 hop reachability). New and existing coordinators are not necessarily neighbors

Table 5.1 Hierarchical, flat, and adaptive routing techniques in WSNs.

Hierarchical routing	**Flat routing**	**Adaptive routing**
Reservation-based scheduling	Contention-based scheduling	Contention-based scheduling
Collisions avoided	Collision overhead present	Collisions overhead present
Reduced duty cycle due to periodic sleeping	Variable duty cycle by controlling sleep time of nodes	Variable duty cycle by controlling sleep time of nodes
Data aggregation by clusterhead	node on multihop path aggregates incoming data from neighbors	intermediate node aggregation
Simple but non-optimal routing	Routing is complex but optimal	Routing is complex and non-optimal
Requires global and local synchronization	Links formed on the fly without synchronization	Links formed on the fly but requires synchronization
Overhead of cluster formation throughout the network	Routes formed only in regions that have data for transmission	overhead of negotiation before transmission
Lower latency as multiple hops network formed by clusterheads always available	Latency in waking up intermediate nodes and setting up the multipaths towards BS	Same as flat-based
Energy dissipation is uniform	Energy dissipation depends on traffic patterns	Energy dissipation depends on application
Energy dissipation cannot be controlled	Energy dissipation depends on implementation	Energy dissipation adapts to traffic pattern
Fair channel allocation	Fairness not guaranteed	Fairness not guaranteed

Table 5.2 Comparison between SPIN, LEACH and Directed Diffusion.

	SPIN	LEACH	Directed Diffusion
Optimal Route	No	No	May be
Network Lifetime	Good	Very Good	Good
Resource Awareness	Yes	Yes	Yes
Use of Meta-Data	Yes	No	Yes

in [118], which, in effect, makes the design less energy efficient because of the need to maintain the positions of two or three hop neighbors in the complicated SPAN algorithm. Table B.1 in Appendix B compares different routing protocols in WSNs using different parameters.

Our scheme, called Grid-based Routing and Aggregator Selection Protocol (GRASP), as a novel data aggregation and routing approach, distinguishes itself from current state of the art solutions in several aspects. First, previous data aggregation schemes perform in-network processing at arbitrary aggregation points (so called opportunistic aggregation), while our scheme finds the set of such points that maximizes the network lifetime. Second, previous schemes solve the problem of selecting the aggregation points and the routing problem as two independent problems, while our approach solves the two problems as one joint problem. To the best of our knowledge, this work is the first to present an exact solution for the joint problem of data routing with aggregation enroute in WSNs. Third, previous schemes utilize dynamic virtual topologies with variable cluster sizes hence incur higher overhead, while our scheme uses a fixed and simple architecture that reduces clustering and routing overhead. Moreover, our routing structure is regular and uniform, which are two desirable features for ad hoc network topologies. Finally, our model is significantly different from other models in the sense that in most of these models, sensor nodes are assumed to generate a constant amount of data to the BS, while our model can handle both constant and variable amounts of data. Furthermore, previous works made the simplistic assumption that an intermediate sensor can aggregate multiple incoming packets into only one single outgoing packet regardless of how many packets are received or the size of the received packets. Our solutions, however, implement a more realistic aggregation process where the aggregation function is dependent on the number and size of incoming packets at each intermediate sensor node.

5.2 Data Routing with Aggregation in WSNs: The Problem Description

We consider a network of energy-constrained sensors that are deployed over a bounded region. Each sensor acquires measurements which are typically correlated with other sensors in its vicinity, and these measurements are to be sent to the BS for evaluation or decision purposes. We assume periodic sensing with the same period for all sensors. During the data gathering process, sensors have the ability to perform in-network processing and data aggregation enroute on the data packets destined to the base station (see Figure 5.1). Given the energy constraints on sensor nodes and the location of these sensors, we are interested in finding an efficient approach by which data is transmitted to the BS using minimal power consumption.

We argue that jointly exploiting the correlation in the data gathered by sensor nodes and optimizing the routing structure in the network can provide substantial energy savings. Therefore, we address the issues of data aggregation methodology, selection of the aggregation points, and optimal routing of aggregated data to the BS together, rather than separately. In other words, we address the joint problem of optimal routing with data aggregation such that the network lifetime is maximized. Indeed, this joint problem is not trivial.

Since data correlation in WSNs is strongest among data signals coming from nodes that are close to each other, the use of a clustering infrastructure will allow nodes that are close to each other to share data before sending it to the BS. Hence, the ideas of fixed cluster-based routing (see chapter 2) together with application-specific data aggregation are used in GRASP to achieve good performance in terms of system lifetime and latency. We show that GRASP improves system lifetime, while incurring acceptable levels of latency in data aggregation, and hence attain the energy and latency efficiency needed for wireless sensor networks without sacrificing quality.

Our scheme, GRASP, consolidates data aggregation and in-network processing at different levels of the virtual grid (see Figure 5.1). In fact, all algorithms presented in this Chapter make use of the rectilinear virtual topology as their underlying transmission architecture. The rectilinear virtual topology is built on top of the physical topology in a manner similar to that of VGA clustering approach (see Chapter 2). However, in this virtual grid, a sensor node i becomes a CH node in a zone with a set of l nodes if the eligibility factor (EF) satisfies the following (refer to system model in chapter 2 for more details):

$$i = arg \max_{j \in l}(EF_j) \quad \text{where} \quad EF_j = a_1 E_j + a_2 e^{-s_j}$$

where E_j is the remaining energy level at sensor node j, while s_j is the number of times a sensor node served as a CH at the time of decision, a_1, a_2 are weighting factors that reflect the importance of each parameter $0 \leq a_2 < a_1 \leq 1$ and $a_1 + a_2 = 1$. The rationale behind the eligibility factor calculated above is that we need to select as a CH the node that has the highest energy reserve and provide fairness in energy draining by allowing all nodes in a zone to participate in the role of CH. The set of CHs form a fixed rectilinear virtual graph G. New CHs, but not new clusters, are chosen at periodic intervals to provide fairness, avoid single node failure, and rotate the energy draining role around the network.

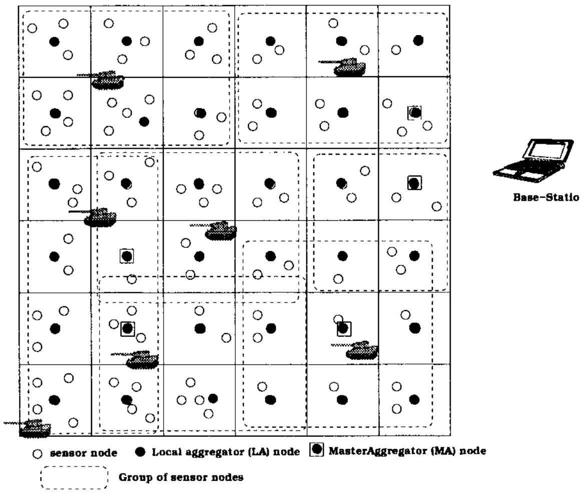


Figure 5.1 Regular shape tessellation applied to the network area with nodes selected to act as aggregators at different levels (sensors are divided into groups).

The set of selected CHs perform two functions. First, they perform the first level of aggregation in their zones, hence they will be referred to as the *Local Aggregators* (LAs) in the rest of the chapter. Second, they relay information towards the BS. To further minimize energy consumption, we propose to use further levels of data aggregation using a subset of LAs. We will introduce algorithms that create a second-level of data aggregation through a subset of LAs called *Master Aggregators* (MAs). We then extend these algorithms to generate a hierarchy (multi-level) of data aggregation points that consolidate data furthermore before being forwarded to the BS. We study the tradeoffs in each case. For a realistic scenario, we assume that a set of LAs which monitor the same phenomenon form a *group*, and the set of LAs is divided into, possibly overlapping, Θ groups. Members of each group, S_g, $1 \leq g \leq \Theta$, are sensing the same phenomenon, and hence their readings are correlated. Hence, each LA node that exists in the overlapping region, will be sending data to its associated MA for each of the groups it belongs to. A group of LAs are assigned one MA and each MA can serve more than one group. We assume that all LAs are aware of the location of the BS. Hence, each node can compute routes to the BS. The location of the BS can be arbitrary and it is used in finding the path(s) from a certain LA node to the BS. An LA node may reach the BS directly if this option results in lower energy cost. Formally, the problem statement is stated as follows:

(a) Joint Routing/MA Selection Problem in two-level scheme (RSP1): *Given a set of LAs, divided into, possibly overlapping, Θ groups, the amount of traffic generated by each LA in each group, an external BS, and a maximum number of Master Aggregators (MAs) equal to M, find a subset of LAs of cardinality p ($p \leq M$) that will act as (MAs) with the objective of maximizing the network lifetime.*

(b) Joint Routing/Aggregation Problem in multi-level scheme (RSP2): *Given a set of LAs divided into, possibly overlapping, Θ groups, the amount of traffic generated by each LA in each group, and an external BS, find a suitable routing scheme and aggregation structure that gather data and report it to the BS such that the network lifetime is maximized. To maximize the network lifetime, the maximum power consumption at each LA node should be minimized.*

Our objective is to minimize the maximum power consumption at all LA nodes in the graph G such that the network lifetime is maximized. To be specific, let P_i be the power consumption by LA node $i \in N$ where N is the set of LAs in the graph G, and let σ be the maximum power consumption among the set of all LA nodes. Then, our objective is to minimize σ, such that

$$\sigma = \max_{i \in N} P_i$$

Note that the network lifetime is inversely proportional to σ, the maximum power consumed by any LA node over the set of all LAs in G. We define the network lifetime as follows:

Definition: The *Network Lifetime* is defined as the time period from the instant when the network starts functioning to the instant when the *first* CH node runs out of energy, i.e., the time until the first zone becomes empty.

Recall that nodes in each zone take turn in serving as CHs according to our clustering scheme. Since CH election is mainly based on power, if a CH fails in a certain zone, then it is guaranteed that this CH is the one with the highest power, and hence all other nodes in the zone have already failed. Note that our definition of lifetime is dependent on the total amount of energy in each zone. We make the assumption that zones have the same energy reserve for reasons that are illustrated later in this Chapter.

5.3 Coverage with Connectivity of VGA: Necessary and Sufficient Conditions

In this section, we derive necessary and sufficient conditions for the VGA to cover the network area as well as remain connected. For the sake of comparison, we follow the notations used in [128]. Let the number of clusterhead (CH) sensor nodes in VGA be n arranged in a rectilinear fashion. Each CH sensor node (or simply CH) fails independently with probability 1-$p(n)$ and is active with probability $p(n)$. Each CH is able to detect events within some distance from it, called the sensing radius $r_s(n)$ and each CH node has a transmission radius, $r_t(n)$. A pair of active CH nodes can communicate directly with each other only if they are at a distance less than or equal to $r_t(n)$. In general, sensing radius is less than transmission radius since sensing radius need to cover a zone only. We show the dependence of the quantities $p(n)$ and $r_t(n)$ on the size of the network n. To simplify notations, we will write p instead of $p(n)$ and r_t and r_s instead of $r_t(n)$ and $r_s(n)$, respectively. We normalize the network area to be a unit square. Two important performance metrics for WSNs are connectivity and coverage.

- **Connectivity of VGA:** VGA is connected if any active CH node can communicate with any other active CH node (possibly using other active CH nodes as relay nodes).
- **Coverage of VGA:** The unit square is said to be covered if every point in the square is within a distance r_s of an active CH node.

In the literature, proofs of connectivity and coverage in WSNs dealt with two cases: use percolation results for planar Poisson placement of nodes over an infinite plane such as [129], or deterministic node placement such as [128] where nodes are manually placed in a form of grid in a unit square. The proofs presented in this section are similar to [128] in the sense that they are constructive in nature, which lead to bounds on the connectivity with coverage. However, our proofs are different from [128] in two aspects. First, proofs in [128] are based on packing the unit square with circular bins, while VGA approach pack the network area with disjoint squares which exactly fits the square network area. Second, our proofs results in equivalent necessary and sufficient conditions, while proofs in [128] results in different bounds for necessary and sufficient conditions due to the extensive packing with overlapping circles in order to ensure coverage in the network area. Unlike [128], the coverage radius in VGA is less than the transmission radius. Moreover, our VGA does not have the border effects which have been neglected in [128]. We now prove necessary and sufficient conditions for coverage, and hence for coverage with connectivity.

Proposition: Consider the VGA clustering approach. Let n be the number of nodes in VGA, with each node being active at time t with probability p. Let $P_s(n)$ be the probability that the network covers the unit square with connectivity. Then, a *necessary* condition for asymptotic connectivity with coverage is given by:

$$pr_t^2 \geq 5\frac{\log(n)}{n} \tag{5.1}$$

Also, for sufficiently large n, we have

$$Pr(\text{network is connected and covered}) \geq 1 - \frac{5}{r_t^2}e^{-\frac{1}{5}npr_t}$$

A *sufficient* condition for asymptotic connectivity with coverage is also given by:

$$pr_t^2 \geq 5\frac{\log(n)}{n}$$

Proof. We divide the unit area into disjoint squares of side length x such that $x = \frac{r_t}{\sqrt{5}}$ as was performed in VGA clustering (see Figure 5.2).

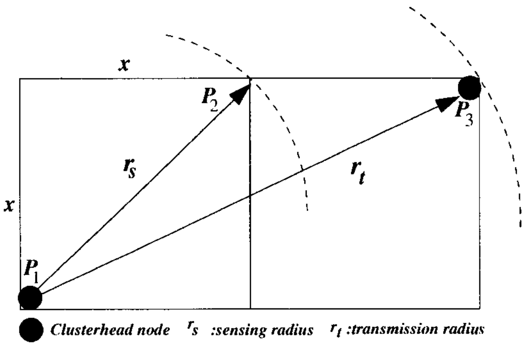

x
P_2
P_3
r_s
r_t
x
P_1
● Clusterhead node r_s :sensing radius r_t :transmission radius

Figure 5.2 The network structure under VGA clustering.

Note that there are a total of $S = \frac{5}{r_t^2}$ small squares in the unit square, labeled as $1, \ldots, S$. We observe that a necessary condition for coverage with connectivity of the unit square is that there should be at least one active CH node anywhere in the square. Let $P_s(n)$ = Pr (there is at least one active CH node in each square). Now, by construction, the squares are disjoint, and in each square of side length x, there are at most $n\frac{r_t^2}{5}$ CH nodes (active or dead). We have counted the number of nodes in each square to be proportional to the area of the square. As each CH node is active with probability p, independent of any other CH node, it follows that

$$\begin{aligned} P_s(n) &= Pr(\bigcap_{j=1}^{S}\{\text{at least one node is active is square } j\}) \\ &= [Pr(\text{at least one active node in square } 1)]^S \\ &= [1-(1-p)^{\frac{r_t^2 n}{5}}]^{\frac{5}{r_t^2}} \end{aligned}$$

Using the fact that $1 - p \leq e^{-p}$, we have

$$\begin{aligned} P_s(n) &\leq [1-e^{-\frac{npr_t^2}{5}}]^{\frac{5}{r_t^2}} \\ &\leq \exp[-\frac{5e^{-\frac{npr_t^2}{5}}}{r_t^2}] \end{aligned}$$

for $P_s(n) \leq 1$ as $n \to \infty$, we need to have

$$r_t^2 e^{\frac{1}{5}npr_t^2} \to \infty \;, \quad \text{as} \quad n \to \infty \tag{5.2}$$

We now define

$$c = \frac{nr_t^2}{\log(n)}$$

Then, from (5.2), it follows that a necessary condition for $P_s(n) \to 1$, is given by

$$c \log(n) n^{cpn-1} \to \infty \;, \quad \text{as} \quad n \to \infty$$

Hence, a necessary condition for coverage with connectivity is that for all n large enough,

$$r_t^2 p \;\geq\; 5\frac{\log(n)}{n}$$

Next, we prove sufficient conditions for coverage with connectivity. We note that a sufficient condition for coverage with connectivity is that there is at least one active CH node in each square of side length x. To see this, consider active CH nodes P_1, P_2, P_3 as shown in Figure 5.2, where P_1, P_3 are active CH nodes within the same square, P_2 is an active node in a neighboring square. The sensing radius r_s in VGA is related to the communication radius by $r_s = \frac{\sqrt{2}}{\sqrt{5}} r_t$. To show that the square tiling of VGA ensures coverage with connectivity, it is sufficient to show that:

$$d(P_1, P_2) \;\leq\; r_s$$
$$d(P_1, P_3) \;\leq\; r_t$$

where $d(.,.)$ is the Euclidean distance. It follows by construction that

$$d^2(P_1, P_2) \leq x^2 + x^2 = 2x^2$$
$$\therefore d^2(P_1, P_2) \leq 2(\frac{r_t}{\sqrt{5}})^2$$
$$d(P_1, P_2) \leq \sqrt{\frac{2}{5}} r_t = r_s$$

For the the connectivity case, it also follows by construction that

$$d^2(P_1, P_3) \leq x^2 + (2x)^2 \leq 5x^2$$
$$d^2(P_1, P_3) \leq 5(\frac{r_t}{\sqrt{5}})^2$$
$$d(P_1, P_3) \leq r_t$$

Let A_i be the event that square i has at least one active node. Let P_c = the probability that there is at least one active CH node in each square. The objective is to find sufficient conditions on r_t, p such that $P_c \to 1$ as $n \to \infty$. Since each CH node is active with probability p, independent of any other CH node, it follows that

$$\begin{aligned} P_c &= Pr(\bigcap_{i=1}^{S} A_i) \\ &= 1 - Pr(\bigcup_{i} \bar{A}_i) \\ &\geq 1 - \sum_{i} Pr(\bar{A}_i) \\ &\geq 1 - \frac{5}{r_t^2}(1-p)^{\frac{1}{5}r_t n} \\ &\geq 1 - \frac{5}{r_t^2}e^{-\frac{1}{5}pr_t n} \end{aligned}$$

Then, for $P_c \to 1$ as $n \to \infty$, we need to have

$$1 - \frac{5}{r_t^2}e^{-\frac{1}{5}pr_t n} < \frac{5}{c}\frac{1}{\log(n)}\frac{1}{n^{c/5-1}}$$

Then for n large enough, we have

$$r_t^2 p \geq 5\frac{\log(n)}{n}$$

□

Our construction allows the control of r_t such that it can always be adjusted to ensure coverage with connectivity. Recall that the relation between the two is governed by the relation:

$$r_s = \frac{\sqrt{2}}{\sqrt{5}}r_t$$

Hence, it is always possible to adjust the value of r_t in the previous results, i.e., adjust the size of the zones in order to maintain the conditions on connectivity with coverage under VGA clustering. Note also that the necessary and sufficient conditions are equivalent under VGA clustering. When compared to [128], where the necessary condition on coverage was

$$r_t^2 p \geq \frac{1}{\pi}\frac{\log(n)}{n}$$

our necessary conditions on coverage with connectivity decreases the condition space, and hence provide more flexible conditions. However, the sufficient conditions for coverage with connectivity derived in [128], which is

$$r_t^2 p > \frac{4}{\pi}\frac{\log(n)}{n}$$

provide better bound than ours at the expense of extensive overlapping between circles to ensure coverage with connectivity in the network area.

In the following sections, we present the constituent algorithms of the GRASP scheme in details.

5.4 Exact Algorithms: An ILP Formulation

In this section, we present exact (optimal) algorithms for both RSP1 and RSP2 problems. For both problems, we introduce an Integer Linear Program formulation to find optimal routes with data aggregation performed enroute to minimize the power consumption, and hence maximize the network lifetime.

5.4.1 RSP1: Two level Routing/Aggregation Problem

In this section, we focus on problem RSP1. We study the interesting problem of selecting the set of MAs such that the network lifetime is maximized. The problem of optimal selection of MAs is NP complete since it is equivalent to the p-center problem in graph theory, which has been shown to be NP complete [52]. Hence, we develop an Integer Linear Program (ILP) that finds the optimal selection of MAs satisfying the aforementioned objective. The objective of the ILP is to select a number of MAs out of the LAs, which does not exceed M, while maximizing the network lifetime. Let us label the base-station as node 0 and label the LA nodes as nodes 1 to n, where n is the total number of LAs. Note that the amount of energy used by a certain node to transmit a message with a certain size is essentially given by the product of (message size×link weight) over the graph G. Those links can be symmetric or asymmetric, and hence have different weights. The link weight represents the amount of power consumed by a node when using that link. The variables and notations used in the ILP are defined in Table 5.3.

Sometimes, only the summarized information is needed to serve the purpose of monitoring environmental events. In this case, the energy used by an MA for transmission depends on the aggregation (summarization) function, which takes different forms including: duplicate suppression, averages, sums, minima, maxima, percentiles, etc. Our ILP can handle different summarization functions[1]. As such, our formulation is generic and can handle any aggregation function and arbitrary traffic streams. As an example, we select the aggregation function to be the *maximum*, i.e., each MA selects the data unit that has the maximum size (length) coming from its constituent LAs source nodes. Now, we present the ILP:

Objective function:

$$Minimize : \alpha * \sigma + \beta * p$$

The objective function minimizes the power consumption at each node while trying to also minimize the number of MAs. The carefully selected values of α and β will ensure that the number of MAs will be reduced only if this does not result in an increase in the power consumption at individual nodes.

[1] Indeed, the ILP can handle any linear function of the data to be aggregated as well as some simple non-linear functions by mapping them to a linear function.

Table 5.3 The ILP variables for problem RSP1.

Notation	Meaning
	Input variables
N	the set of LAs in the graph G; with cardinality $n{=}\lvert N\rvert$ indexed by $1 \le i \le n$.
M	the maximum number of MAs that can be allocated.
P_{max}	the maximum power available for each LA node.
α, β	two weighting numbers (scalars) and $\alpha \gg \beta$.
Θ	Total number of groups.
S_g	The set of LA nodes of group g, $1 \le g \le \Theta$.
m_i^g	The size (number of traffic units) of the packet sent by LA node $i \in S_g$.
Q	A very large number such that ($Q > \max_{g,1\le i\le n}(m_i^g)$)
σ	the maximum power level among the set of all LA nodes.
	Variables determined by the ILP
I_{ij}^g	A binary variable which is 1 iff LA node i of group g uses LA node j as its MA.
p	number of MAs allocated by the ILP ($p \le M$).
Z_j^g	the maximum number of data units (packets) received by the jth MA node from members of group g, where $0 \le Z_j^g \le \max_{i,I_{ij}^g=1}(m_i^g)$
X_j^g	an auxiliary variable such that $X_j^g = I_{jj}^g * Z_j^g$, hence $0 \le X_j^g \le \max_{i,g}(m_i^g)$
R_{ij}^a	a binary indicator which is 1 if and only if LA node a is on the route from node i to node j.
F_{ij}	A binary indicator which is 1 if and only if nodes i and j are connected by a link.
Y_{lk}^{ij}	a binary indicator; is 1 if and only if the traffic stream sourced at i and destined to j uses the link between node l and node k.

Subject to:

$$\sum_{j=1}^{n} I_{ij}^g = 1 \text{ ,} \forall g \text{ ,} i \in S_g \tag{5.3}$$

$$\sum_{g} \sum_{j \in S_g} I_{jj}^g = p \tag{5.4}$$

$$p \le M \tag{5.5}$$

$$I_{ij}^g - I_{jj}^g \le 0 \text{ ,} \forall \quad 1 \le j \le n,\ g,\ i \in S_g \tag{5.6}$$

$$Z_j^g \ge m_i^g I_{ij}^g \text{ ,} \forall\, 1 \le j \le n, 1 \le g \le \Theta, i \in S_g \tag{5.7}$$

$$X_j^g \ge Q I_{jj}^g - Q + Z_j^g \text{ ,} 1 \le g \le \Theta, 1 \le j \le n \tag{5.8}$$

$$X_j^g \le Z_j^g \text{ ,} 1 \le g \le \Theta, 1 \le j \le n \tag{5.9}$$

$$\sigma \ge P_i \text{ ,} \forall\, i \tag{5.10}$$

$$\sigma \leq P_{max} \tag{5.11}$$

Constraint (5.3) ensures that each LA node of group g is associated with only one MA node. Constraint (5.4) ensures that total number of allocated MAs for all groups is p. Constraint (5.5) guarantees that the number of MAs cannot exceed M. Constraint (5.6) ensures that an LA node i of group g will use LA node j as its MA only if LA j has been selected to act as an MA j. Constraint (5.7) ensures that the maximum-sized packets are sent to the BS taken over all the packets generated by the LAs of the groups served by the jth MA. Constraints (5.8) and (5.9) (together with the minimization of the objective function) set X_j^g to the amount of traffic routed from the jth MA to the BS, i.e., the product $I_{jj}^g * Z_{jj}^g$. Constraint (5.10) finds the maximum power level over all LA nodes. Constraint (5.11) guarantees a minimum node lifetime and limits the maximum power consumption of any node, i.e., limits sensor participation in data communication in the network. To show that our ILP formulation can be easily modified to handle other aggregation functions, consider, for example, the aggregation function to be duplicate suppression, where only one copy of data received at each MA is sent to the BS. Then, the ILP will be a special case of the above ILP where the variable m_i^g will be replaced by a constant value, say k^g, which is the size of the first packet received, and the three constraints (5.7)-(5.9) are replaced by the single constraint,

$$X_j^g = k^g I_{jj}^g \quad ,1 \leq g \leq \Theta, 1 \leq j \leq n$$

We assume that each zone has a finite energy reserve and that each zone will have the same amount of energy (E), which is measured in joules. The lifetime of a zone i, denoted by t_i, is obtained as $t_i = E/P_i$, where P_i is the average power consumed in zone i in watts. Hence, the maximum value of P_i must be minimized in order for the zone i lifetime, t_i, to be maximized. We make the assumption that zones have the same energy reserve for two reasons. First, if the network is dense, each zone will have approximately the same number of nodes, and hence it will roughly have the same amount of energy reserve. Even if the nodes start with different energy levels, we assume that random distribution of heterogeneous nodes will result in the same energy reserve in each zone. Secondly, if the zones have different values of E, the problem will be non-linear rather than linear, and hence much harder to formulate and solve. Therefore, we assume equal values of E in order to keep the problem manageable.

The power consumption is dependent on the determination of the actual routing of the data traffic, i.e., we must find the power consumed by each LA source node when participating in routing data over G. The following additional set of constraints are required for performing route computations needed to find the values of R_{ij}^a, which is defined as a binary indicator that has a value of 1 if and only if LA node a is on the route from node i to node j. Let Y_{lk}^{ij} be a binary indicator and has a value of 1 if and only if the traffic stream sourced at i and destined to j uses the link between node l and node k. The following two constraints ensure that for the connection going from node i to node j, no traffic is coming in (going out) the source i (destination j), respectively

$$\sum_{k,F_{ki}=1,k\neq i} Y_{ki}^{ij} = 0 \quad \forall i,j \ ; \quad \sum_{k,F_{jk}=1,k\neq i} Y_{jk}^{ij} = 0 \quad \forall i,j$$

The following two constraints ensure that the connection traffic between i and j is originating(terminating) at $i(j)$, respectively

$$\sum_{k,if F_{ik}=1,k\neq i} Y_{ik}^{ij} = 1 \quad \forall i,j \ ; \quad \sum_{k,if F_{kj}=1,k\neq j} Y_{kj}^{ij} = 1 \quad \forall i,j$$

The following constraint preserves the continuity of connection traffic on one of multiple possible routes

$$\sum_{k,if F_{kx}=1,k\neq x,k\neq j} Y_{kx}^{i,j} = \sum_{j,if F_{xk}=1,j\neq x,j\neq s} Y_{xk}^{i,j} \quad \forall x,(x\neq i,j),i,j \tag{5.12}$$

The following two constraints determine the value of R_{ij}^{a}, i.e., if sensor node a is on the route(s) between nodes i and j.

$$R_{ij}^{a} \geq \sum_{k} Y_{ka}^{i,j}/Q \quad \forall\ i,j,a,i\neq j,i\neq a,j\neq a \tag{5.13}$$

$$R_{ij}^{a} \leq \sum_{k} Y_{ka}^{ij} \quad \forall\ i,j,a,i\neq j,i\neq a,j\neq a \tag{5.14}$$

The power consumed by node a is given by

$$P_a \quad = \quad \sum_{j,g=1} I_{aj}^{g} m_f^{g} \quad + \quad \sum_{i,j,g} I_{ij}^{g} R_{ij}^{a} m_a^{i} \quad + \quad \sum_{g=1}^{G} \Theta I_{aa}^{g} X_f^{g} \quad + \quad \sum_{j,g,j\neq a} I_{jj}^{g} X_j^{g} R_{j0}^{a}$$

Where the first term refers to the power consumed by data sourced out from LA node a, the second term refers to power consumed by node a while relaying for other LAs towards an MA node; the third term refers to the power consumed by data sourced out from MA node a; and the last term refers to power consumed while relaying for other LAs towards the BS. Let $U_{ij}^{ga} = I_{ij}^{g} \bigwedge R_{ij}^{a}$ and let $V_{jj}^{g} = I_{jj}^{g} * X_j^{g}$. Note that the product $I_{jj}^{g} * X_j^{g}$ is non-linear. We map the non-linear term to a linear representation by using the following two linear equations.

$$V_{jj}^{g} \geq QI_{jj}^{g} - Q + X_j^{g} \quad ; \ V_{jj}^{g} \leq X_j^{g} \tag{5.15}$$

Since in the objective function the total energy cost will be minimized, this will force the variable V_{jj}^{g} to take the smallest feasible value within the limits specified by above two equations, thus providing an exact product of $I_{jj}^{g} * X_j^{g}$. As for U_{ij}^{ga}, the following two constraints perform the conjunction of I_{ij}^{g} and R_{ij}^{a}:

$$U_{ij}^{ga} \geq I_{ij}^{g} + R_{ij}^{a} - 1 \quad ; \ U_{ij}^{ga} \leq (I_{ij}^{g} + R_{ij}^{a})/2 \tag{5.16}$$

Let us define $W_{jj}^{ga} = V_{jj}^{g} * R_{j0}^{a}$. Again, we map this non-linear term using the following two linear equations.

$$W_{jj}^{ga} \geq QR_{j0}^{a} - Q + V_{jj}^{g} \quad ; \ W_{jj}^{gn} \leq V_{jj}^{g} \tag{5.17}$$

Manipulating the previous linear equations, the power consumed on an LA node a when it participates in routing between nodes i and j can now be given as:

$$P_a \quad = \quad \sum_{g=1,a\in S_g} \sum_{j=1,j\neq a}^{n} I^g_{aj} m^g_a \; + \sum_{i,j,g,i\neq a,j\neq a,i\neq j,i\in S_g} U^{g,a}_{ij} m^i_g \; + \sum_g V^g_{aa} \; + \sum_{g,j,(j\neq a)} W^{g,a}_{jj} \qquad (5.18)$$

Since the ILP can solve small to medium sized networks, the optimal solution obtained from the ILP is used as a baseline against which heuristic approaches presented in Section 5.5 will be compared. Due to their quick convergence, these heuristics are used in large sensor networks. The ILP can also be used to assess the optimality of any other heuristic approach.

5.4.2 RSP2: Multi-level Routing/Aggregation Problem

In this section, we present exact algorithm for the RSP2 problem. We extend the ILP presented in the previous section to handle multiple levels of data aggregation[2]. We anticipate that multilevel aggregation will result in further reduction of energy dissipation, and hence prolong the lifetime more than the two-level data aggregation scheme. Note that each intermediate sensor node, that is acting as LA, can also act as an aggregator for data coming from a subset of LAs that belong to the same group in addition to acting as an LA for the sensors in its own zone. An example of a sensor network that performs routing with multi-level data aggregation is shown in Figure 5.3. There are seven source nodes reporting data back to the BS (or sink) node using multiple aggregation points enroute. Our objective is to find optimal routes and a set of aggregation points for each group such that the network lifetime is maximized. The problem of optimal selection of multiple levels of aggregation is harder than the two level aggregation scheme, which has been shown to be NP complete and with no regards to groups [52]. Hence, we approach this problem by developing an Integer Linear Program (ILP) with the objective of performing optimal routing with multi-level data aggregation in order to maximize the network lifetime. The ILP formulation presented in this section is generic and can handle arbitrary data aggregation factors and arbitrary traffic streams. The variables and notations used in the ILP are defined in Table 5.4. Some of these notations already appeared in Table 5.3 and are repeated here for easy reference.

We define r^g_k as the fraction of original traffic of a source left after aggregation, when data streams from k members of group g are aggregated. Hence the data aggregation factor is $(1 - r^g_k)$. Now, we present the ILP:

Objective function:

$$\text{Minimize } \sigma$$

The objective function minimizes the maximum power consumption at each node.

[2] The aggregated capacity computation part in the ILP presented in this section is an adaptation of an ILP developed for the design of optical networks with many-to-one traffic grooming [120].

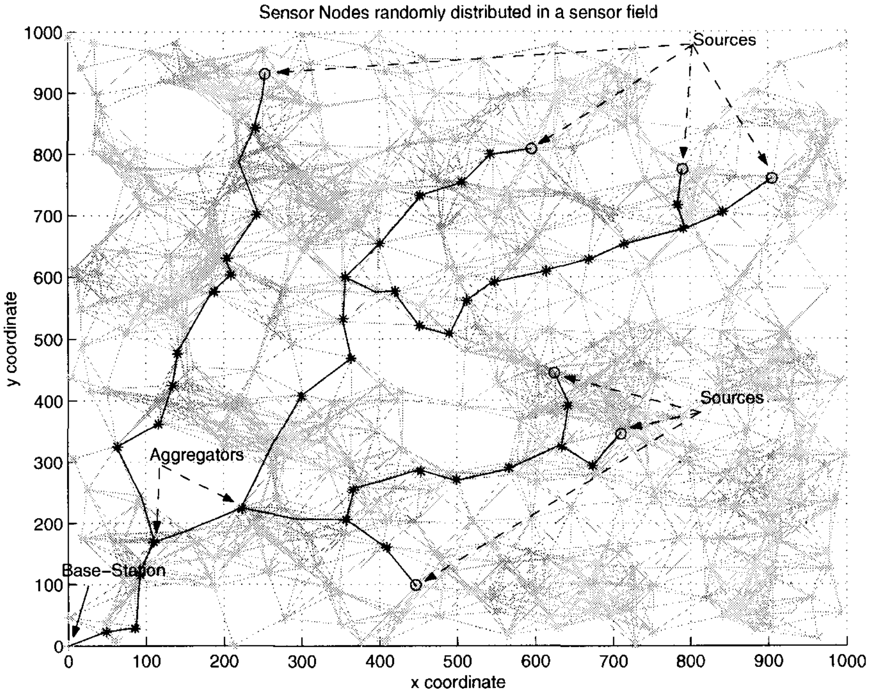


Figure 5.3 An example of multi-level data aggregation tree with number of LA source nodes = 7.

Subject to:

The following two constraints find the maximum power consumption over all LA nodes, while respecting the maximum power consumption limit of any LA node,

$$\sigma \geq P_i \quad ,\forall\, i \tag{5.19}$$

$$\sigma \leq P_{max} \tag{5.20}$$

The following constraint ensures that the power consumed at each LA node is sufficient to send the amount of required traffic to the other nodes.

$$P_i \geq \sum_{g} \sum_{j\ ,\text{if}\ F_{ij}=1} X_{ij}^{g} * w \ ,\forall g\ ,i \tag{5.21}$$

where w is a constant that represents the amount of power spent per byte (watts/byte). The following two constraints ensure that if k is the number of members of group g sending data from node i to node j, then the value of $I_{ij}^{g,k}$ is 1 for all the values of k that are less than or equal to the actual number of traffic streams at node i; otherwise it will be 0.

$$\sum_{k=0}^{|S_g|} I_{ij}^{g,k} + \sum_{s\in S_g} T_{ij}^{s,g} = |S_g| + 1 \ ,\forall g\ ,j\ ,i,\ \text{if}\ F_{ij} = 1 \tag{5.22}$$

$$I_{ij}^{g,k} \leq I_{ij}^{g,k+1}\ ,\ \forall g,\ 0 \leq k \leq |S_g| - 1, i, j,\ \text{if}\ F_{ij} = 1 \tag{5.23}$$

Table 5.4 The ILP variables for problem RSP2

Notation	Meaning
	Input variables
N	the set of LAs in the graph G; with cardinality $n{=}\|N\|$ indexed by $1 \leq i \leq n$.
Θ	total number of LA groups in G.
S_g	set of LA nodes of group g, $1 \leq g \leq \Theta$.
m_i^g	number of traffic units (e.g., packets) sent by LA node $i \in S_g$.
r_k^g	fraction of original traffic of a source left after aggregation, when data streams from k members of group g are aggregated such that $r_k^g \geq r_{k+1}^g$ and r_k^g*k$\leq$ $r_{k+1}^g{}^*k + 1$, $0 \leq k \leq \|S_g\|$ with $r_1^g = 1$ and $r_0^g = 0$.
P_{max}	the maximum power available for each LA node.
F_{ij}	a binary indicator which is 1 if and only if nodes i and j are connected by a physical link.
Q	A very large number such that $(Q > \|S_g\| \max_{g,1 \leq i \leq n}(m_i^g))$
w	A constant which represents the amount of power spent per byte (watts/byte).
$e_{i,j}$	the link weight between LA nodes i and j, which is assumed *symmetric* in graph G.
	Variables determined by the ILP
$X_{i,j}^g$	number of data units coming from members of group g and sent by node i to node j after aggregation.
P_i	the amount of power utilized at LA node i.
$T_{ij}^{s,g}$	a binary indicator; is 1 if and only if the traffic stream sourced at node s of group g uses the link between nodes i and node j.
$I_{ij}^{g,k}$	a binary indicator; is 1 if and only if the number of LA source nodes in group g that are sending data through the link from node i to node j is $\leq k$.
M_{ij}^g	a binary indicator; is 1 if and only if the link going from node i to node j carries the traffic by at least one source of group g.

The following two constraints together give the exact amount of traffic sent by node i to node j, once node i has aggregated the data coming from members of group g:

$$X_{ij}^g \geq I_{ij}^{g,k} * r_k^g * \sum_{s \in S_g} m_s^g T_{ij}^{s,g} \ ,\forall g\ ,j\ ,k\ ,i, \text{ if } \ F_{ij} = 1 \tag{5.24}$$

$$X_{ij}^g \leq r_k^g * \sum_{s \in S_g} m_s^g T_{ij}^{s,g} + Q * I_{ij}^{g,k}(k - \sum_{s \in S_g} T_{ij}^{s,g}) \ ,\forall g\ ,j\ ,k\ ,i, \text{ if } \ F_{ij} = 1 \tag{5.25}$$

However, constraint(5.24) is nonlinear, but it can be linearized as follows

$$X_{ij}^g \geq Q * I_{ij}^{g,k} - Q + r_k^g * \sum_{s \in S_g} m_s^g T_{ij}^{s,g} \ ,\forall g\ ,j\ ,k\ ,i, \text{ if } \ F_{ij} = 1 \tag{5.26}$$

Also constraint(5.25) is non-linear, and can be linearized as follows

$$X_{ij}^g \leq r_k^g * \sum_{s \in S_g} m_s^g T_{ij}^{s,g} + Q|S_g| * (1 - I_{ij}^{g,k}) + Q * k * I_{ij}^{g,k} - Q * \sum_{s \in S_g} T_{ij}^{s,g} \ ,\forall g\ ,j\ ,k\ ,i, \text{ if } \ F_{ij} = 1 \tag{5.27}$$

The following three constraints ensure that the traffic streams once aggregated will not be split on the way to the BS:

$$M_{ij}^{g} \geq \sum_{s \in S_g} T_{ij}^{s,g}/Q \ , \ \forall g, \ ,i, \ j, \text{ if } F_{ij} = 1 \tag{5.28}$$

$$M_{ij}^{g} \leq \sum_{s \in S_g} T_{ij}^{s,g} \ , \ \forall g, \ ,i, \ j, \text{ if } F_{ij} = 1 \tag{5.29}$$

$$\sum_{j, \text{ if } F_{ij}=1} M_{ij}^{g} \leq 1 \ , \ \forall g \ ,i \tag{5.30}$$

The guarantee of a minimum lifetime of an LA node is highly dependent on the determination of the actual routing of the data traffic, i.e., we must find the power consumed by each LA source node when participating in routing data over G. The following additional set of constraints are required for performing route computations.

The following two constraints ensure that for the connection going from node i to node j, no traffic is coming in (going out) the source i (destination j), respectively

$$\sum_{i,\text{if } F_{is}=1, s \neq i} T_{is}^{s,g} = 0 \quad ; \quad \sum_{j,\text{if } F_{0j}=1, j \neq 0} T_{0j}^{s,g} = 0 \quad \forall \, g, \ s \in S_g$$

The following two constraints ensure that the connection traffic between i and j is originating (terminating) at i (j), respectively

$$\sum_{j,\text{if } F_{sj}=1, s \neq j} T_{sj}^{s,g} = 1 \quad ; \quad \sum_{i,\text{if } F_{i0}=1, s \neq 0} T_{i0}^{s,g} = 1 \quad \forall \, g, \ s \in S_g$$

The following constraint preserves the continuity of connection traffic on one of multiple possible routes

$$\sum_{i,\text{if } F_{ix}=1, i \neq x, s \neq k} T_{ix}^{s,g} \ = \ \sum_{j,\text{if } F_{xj}=1, j \neq x, j \neq s} T_{xj}^{s,g} \quad \forall \, x, g, \ s \in S_g, \ (1 \leq x \leq n, \ x \neq s) \tag{5.31}$$

5.5 Heuristic Approach

In the previous section, we presented an optimal solution for both **RSP1** and **RSP2** problems using an ILP formulation. In practice, WSNs are large networks with hundreds of nodes, which the ILPs cannot handle with feasible computation time. Therefore, a more efficient, albeit less optimal, scheme is highly desirable. In this section, we present several heuristics to find near optimal solutions for both **RSP1** and **RSP2** problems. Specifically, we present four approximate algorithms. The first two algorithms are based on the genetics algorithms approach [123]. The other two algorithms called, Load Balancing with Aggregation (LBA) and Clustering-Based Aggregation Heuristic (CBAH) algorithms, aim at balancing power consumption in order to prolong the network lifetime. We have also designed and experimented with other approximate algorithms, e.g., a modified k-means clustering algorithm [124] and a simple greedy algorithm for the two-level aggregation scheme [119]. However, we will neither present the details nor the results for k-means and the greed approach since they are inferior to the four heuristics presented in this section.

5.5.1 A Genetic Algorithms Approach

Genetic Algorithms (GA) [123] is a general combinatorial search technique. The major step in GA is to find an efficient way to represent the solution. The detailed operation of GA can be found in [123]. To streamline our presentation, Figure 5.4 shows a high level description of the GA operations. In the GA algorithm, T is the population of individuals (chromosomes) and T_{best} is the best individual that has ever existed, which constitutes the fittest individual.

```
Algorithm GA:
Create-initial population ($T$);
$T_{best}$=best-individual ($T$);
for (generation=1 until no-change-in-fitness) do
    $T_n$=0;
    $T_1$=selection($T$);
    $T_2$=selection($T$);
    $T_n$=$T_n \cup$ crossover($T_1$,$T_2$);
    fitness-calculation($T_n$);
    $T$=reduction($T \cup T_n$);
    $T_{best}$=best-individual($T_{best} \cup T$);
    mutation($T$);
    fitness-calculation($T$);
end
```

Figure 5.4 A high level description of the GA Algorithm.

5.5.1.1 GA for RSP1

Figure 5.5 shows how we represent our solution for the MAs selection problem using a binary string of length n. A value of 1 in the ith cell in the binary string indicates that node i acts as an MA node; otherwise it is an LA node. The quality of this solution structure is evaluated to produce a measurement of the individual's (chromosome) fitness. First, we generate an initial population of randomly created, and thus different, MAs structure. After determining the fitness of each individual solution, we reproduce solutions based on fitness, giving more chances to fitter individuals (solutions) until a stopping criterion is met. Our stopping criterion corresponds to encountering a fixed number of consecutive generations for which the solution does not improve. For selection, we used a roulette wheel. Figure 5.5(b),(c), and (d) shows the process of reproducing generations, crossover, and mutation, respectively. In the same figure, we also show how a complete generation may interact and produce a

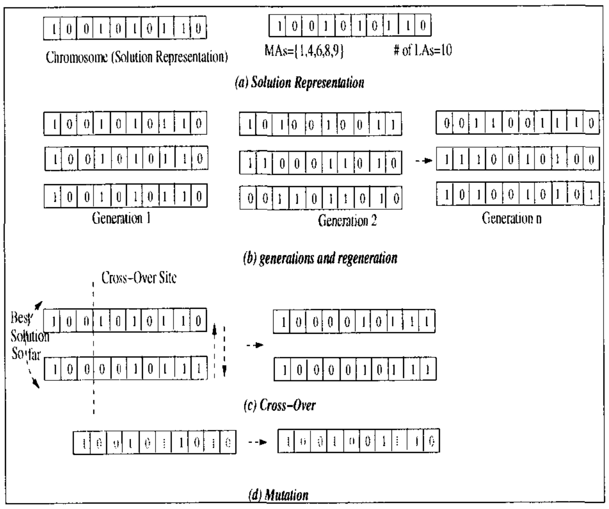


Figure 5.5 MAs selection: representation of solution in the form of a binary string, for n=10 and p=5.

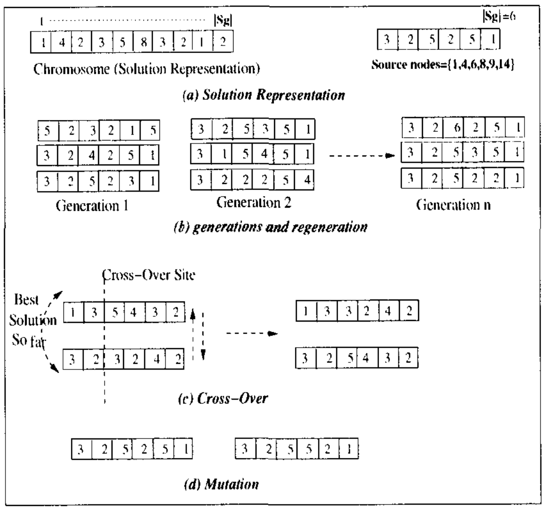

|Sg|
|Sg|=6
Chromosome (Solution Representation)
Source nodes={1,4,6,8,9,14}
(a) Solution Representation
Generation 1
Generation 2
Generation n
(b) generations and regeneration
Cross-Over Site
Best Solution So far
(c) Cross-Over
(d) Mutation

Figure 5.6 representation of solution in multi-level aggregation scheme in the form of routing structure.

new generation through the application of crossover and mutation. We select a crossover site randomly and then exchange the left part of the chromosome with the right one. The mutation operator is applied with a very small probability and operates on an individual chromosome. For mutation, we randomly pick two cells in a chromosome such that one of them has a value of one and the other is zero. After that we just swap their values. The number of individuals is kept constant throughout all generations.

5.5.1.2 GA for RSP2

The GA operation for RSP2 follows the same operational steps of that developed for RSP1 except for the solution representation. We assume that for each source s in group $g \in \Theta$, the number of available routes, $R(s, BS)$ to the BS is known. The route discovery can be implemented by reactive techniques, which might return large number of routes between s and the BS. However, each source will only use a limited number of the available routes for the sake of routing with aggregation. Denote by R^s_{max} the set of these limited number of routes between $s \in S_g$ and the BS. Also, indicate by $N(r_s)$ the number of nodes on route r_s, $r_s \in R^s_{max}$. An efficient way to represent the solution (*individual*) for RSP2 is as follows. For each group $g \in \Theta$, the solution is represented as a string of length $|S_g|$, where $|S_g|$ is the number of LAs in group g. The ith cell in the string contains the route number that will be used by the ith source of group g, which has an integer value between 0 and R^i_{max} (see Figure 5.6 for solution representation). Hence, the *individual* (chromosome) is a routing structure for each group. The quality of this routing structure is evaluated to produce a measurement of the individual's fitness. First, we generate an initial population of randomly created, and thus different, routing structure for each group g. The generated population is subjected to a simulated evolution process consisting of the three main components, namely, selection, crossover, and mutation as was discussed in the GA for RSP1. Figure 5.6(b),(c), and (d) shows the process of reproducing generations, crossover, and mutation, respectively. Since the mutation operator has access to all individuals, the best individual is saved in each generation before the mutation operator is applied. At the end of the algorithm, the best individual, T_{best}, that has ever existed constitutes our final routing with aggregation solution.

We wrote a program to solve the proposed genetics algorithm. Our program finds a set of routes and a set of aggregators on those routes for the sake of maximizing the network lifetime. Note that the number of aggregation points used by a group depends on the selected route by each source node s in that group. To find the set of aggregation points for a group, our program inspects the set of routes used by each source of that group and determine the nodes that are common between these routes. These common nodes are considered candidate aggregation points for the data coming from source nodes of that group. However, since aggregated data cannot be split after leaving the aggregation point, the program unify the rest of the route for the aggregated data unless further source nodes of that group join at some point on the unified route. In this case, the data is fused and the rest of the path is unified again and so on until the base-station is reached. The rest of the route was selected as the longest remaining route of participating nodes to allow for further aggregation. By knowing how many levels of aggregation, we can determine the level of aggregation the raw data of each source node in that group was exposed to during its journey to the BS.

5.5.2 Balanced Power Consumption Heuristics

In this section, we present two heuristics that balance the power consumption at sensor nodes for the sake of maximizing the network lifetime in both two- and multi-level data aggregation schemes.

5.5.2.1 LBA: Load Balancing with Aggregation Heuristic for RSP1

The first heuristic finds a (near) optimal solution for the RSP1 problem. The motivation behind this heuristic is inspired by some observations taken from our ILP developed in Section 5.4.1. After running the ILP for many scenarios, we noticed that the power consumed by different LA nodes is almost always the same. In other words, if we define

$$\Delta = \max_{i,j}(|P_i - P_j|) \ , \forall\, i,j \in N, i \neq j$$

as the maximum absolute difference in the power consumption levels between any two LA node pair (i, j) in the graph G, then the value of Δ should almost always equal to zero. This indicates that the optimal solution can achieve balanced power consumption among all LA nodes in our scheme. Therefore, we make use of this fundamental observation to develop a simple and efficient heuristic, which can be used to solve for large WSNs. The heuristic is called LBA: Load Balancing with Aggregation Algorithm. In the virtual graph G, LA nodes are numbered sequentially from 1 to n starting from the left-upper corner and starting from 1 and then from left to right proceeding row by row. The LBA algorithm starts with the first node in G and proceeds sequentially through the whole topology in a left-to-right and top-down fashion. Our objective is to compute a set of MAs out of the set of LAs (groups) that achieves balanced power consumption, and hence maximize the network lifetime.

```
Data      : $N$, $\Theta$, $M$, $m_i$.
Result    : A set of routes and a set of $M$ MAs.
Algorithm LBA:
initialization: Compute $W$, allocated_MAs[]=$\phi$;
while ($M$!=0) do
    for ($i$=1 to $\Theta$) do
        set $C[i]$=0; min_powr=$\infty$;
        for ($j$=1 to $n$) do
            X=COMPUTE-POWER($S_i$,$j$);
            X=X+ power-consumed($j$,0);
            if (X < min_powr) then
                min_powr=X;
                index=$j$;
            end
        end
        if (min_pwr < $W$) then
            allocated_MAs[]=allocated_MAs[] $\bigcup$ index;
            $h$=LOOKUP(allocated_MAs[]);
            if (match && X+$C[h]$ < $W$) then
                $MA_i$=$h$;
                $C[h]$=$C[h]$+X;
            end
        else
            $MA_i$=index;
            $C[i]$=min_powr;
            $M$=$M$-1;
        end
        // allocate MAs for the remaining SS's;
        if ($M$==0 && $i < \Theta$) then
            for ($k$=1 to ($\Theta$-$i$)) do
                least_capacity=$\infty$;
                for ($l$=1 to $M$) do
                    if ($C[l]$ < least_capacity) then
                        least_capacity=$C[l]$;
                        selected_MA=$l$;
                    end
                end
            end
            ASSIGN-MA($k$,$l$);
        end
    end
end
```

Figure 5.7 A high level description of the LBA heuristic for the RSP1 problem.

Figure 5.7 gives a high level description of LBA. LBA takes as an input N: set of LA sensor nodes (cardinality $n = |N|$ with a base-station node labeled node 0, Θ, number of LA sensor groups, M, number of MAs to be allocated, and m_i, the amount of traffic sent by the ith LA node, i=1,...,n. We refer to each group g, $1 \leq g \leq \Theta$ as a *supersensor* (SS). Each SS will be associated with an MA node. However, the same MA node can serve more than one SS. We find the value of the initial power, W, that each LA, acting as an MA, will be using to send data to the BS. This value is then used in selecting the set of MAs and the LAs that use them given all the traffic units collected by an LA node from its constituent members and based on the aggregation function. Consider the *maximum* size of the generated packets to be the aggregation function, then W will be computed as the sum of the maximum traffic units coming from each SS taken over the set of all super-sensors, and then divided by the number of the allocated MAs, namely, M. Thus,

$$W = \sum_{S_g, g=1,\ldots,\Theta} (\max_{j \in S_g}(m_j))/M$$

Next, the assignment of MAs to the set of super-sensors takes place. We observed that the MA assignment problem is similar to the classical bin-packing problem, but with a major difference: the identities of the bins (MAs) are unknown and the power required by each LA can be different for different MAs. In other words, we have a total of M bins, not known in advance, each of capacity W. We want to assign one or more super-sensors (SS) to an LA node that can act as an MA node such that the total power coming out of this LA is W and the total number of allocated LAs does not exceed M. Note that standard bin-packing solutions may not work, leading to original bin-packing problem and original MA selection problem, which depends on the allocation criterion that will be used.

In LBA, the set of M MAs are selected based on incremental filling of the W capacities. Starting from the first SS, the algorithm iterates over all LA nodes to compute the power consumption needed to reach BS through each of these LA nodes. The function COMPUTE-POWER(S_i,j) will compute the power consumption for SS $S_i, i = 1, ..., \Theta$ to LA node $j = 1, ..., n$. The first SS will select an LA node as its MA node if the power out of this LA is less than W and it is the least among the set of all LAs. At each selected LA that act as an MA, LBA memorizes the value of the fraction of the allocated power so far. LBA stores this allocated fraction power in a variable $C[i], i = 1, ..., M$. Similarly, the algorithm computes for the second SS the power consumption to each LA node and will select the the LA node with minimum value of power consumption as the MA to this SS. If it happens that the selected LA node is the same one selected by a previous SS, LBA checks to see if the selected LA node can accommodate both SS's, i.e., the addition of its power consumption to $C[1]$ is less than $W + \delta$, where δ is small number. If not, the LA node with the second least power consumption is selected as MA. The function LOOKUP(allocated-MAs[]) will search previously allocated LAs for a possible match with current selected LA node. The process continues until all allocated MAs are associated with at least one SS and the total number of allocated MAs does not exceed M. If the number of available MAs, M, is exhausted before all SSs are covered, the remaining SSs are forced to select an already allocated MA starting from the least occupied one. The function ASSIGN-MA(k,l) assigns the SS number l to the least occupied and already allocated kth MA node. Note that it may happen sometimes that we need to over-provision some of the already allocated MAs, and we limit the amount of over-provisioning to δ. The algorithm LBA has a time complexity on the order of $O(\Theta * M * n + \Theta * M)$ to compute the set of M MAs. In the worst case, the algorithm takes $O(n^2 + M * n)$ to finish. As for the space complexity, the LBA algorithm need a space on the order of $O(n) + O(\Theta) + O(M) = O(n)$.

5.5.2.2 CBAH: Clustering-Based Aggregation Heuristic for RSP2

In this section, we present a heuristic to find a (near optimal) solution for the **RSP2** problem. The motivation is to balance the power consumed by different LA nodes. We now describe the algorithm, called Clustering-Based Aggregation Heuristic (CBAH). In order to achieve balanced power consumption among LA nodes, CBAH will use the information about the total power consumption experienced at each LA node so far in addition to the required transmission power[3]. Note that a node with high total power consumption or low remaining energy should be avoided when performing routing, hence prolonging its lifetime. Let P_i^t denote the total power consumption at LA node i measured from the instance the network starts functioning. CBAH will find routes on G for each source node of each group such that power consumption in the network is balanced. In order to achieve balanced power consumption and extend network lifetime, a source node should select a route that minimizes the maximum total power consumption (P_i^t) at each individual node i in the route taken over all feasible routes for that source node, while allowing for data aggregation enroute.

Each source node s in each group $g \in \Theta$ can have many disjoint and non-disjoint routes to the destination (BS). However, it is practical to settle for routing through a limited number of *semi-disjoint* paths at a time. We thus consider the problem of finding a limited number of semi-disjoint routes R, where R is a small fixed integer. Let $\mathcal{R}$ be the set of these semi-disjoint paths. In fact, any source node can have a maximum of four semi-disjoint routes to the BS as communication is performed in vertical-horizontal directions only on the virtual graph G. Denote by $N(r_s)$ the set of nodes on route r_s, $r_s \in \mathcal{R}$ for source node s. CBAH will select for each source node s of each group $g \in \Theta$ a route (r) among all possible $\mathcal{R}$ routes such that,

$$r = \arg\min_{r_s \in \mathcal{R}} \{ \max_{i \in N(r)} (P_i^t) \}$$

The decision of selecting such a route has the objective of maximizing the lifetime of each node on the route through balancing the power consumption in the network. In addition, this criterion smoothes the use of the battery of each node and extends its lifetime.

Figure 5.8 gives a high level description of CBAH. CBAH takes as an input the set of groups Θ, group members S_g, the traffic (m_g^i) generated at each source node $i \in S_g$ of each group $g \in \Theta$, and the

[3] This also applies to the case when sensor nodes have different initial energy levels (heterogeneous network). In this case, the remaining energy at each node can be used.

```
Input: $A$, $n$, $m_i^g$, $\Theta$, $S_g$;
Output: A set of routes and aggregation points to maximize the network lifetime.;
Algorithm CBAH:
Initialize;
Label all LA nodes as "un-visited";
Set $P_i^t$=0.0; $1 \leq i \leq n$, aggregator[$g$][$n$]={$\phi$}, $N_r^s$={$\phi$}, routes[$g$][$s$]={$\phi$}, $R_{max}$=4, route-found=false, num-sources=0;;
Define set of struct candidate-list[], new-candidate-list[], nbr, element, path-element;
for ($g$=1 to $\Theta$) do
    for ($j$=1 to $|S_g|$) do
        $r$=1 /*for each source node $j$*/;
        while ($r \leq \mathcal{R}$) do
            $s$=get-ID($j$);
            $N_r^s$=$N_r^s \cup s$;
            candidate-list[]={$\phi$};
            new-candidate-list[]={$s$};
            while ($0 \not\subseteq$ new-candidate-list) do
                /*BS is not reached*/;
                candidate-list $\leftarrow$ new-candidate-list;
                new-candidate-list={$\phi$};
                for (element $\in$ candidate-list) do
                    for (each nbr of element) do
                        /* nbr is obtained from $A$*/;
                        /*The nbr node is located above, below, at the left or at the right of a node.*/;
                        if (nbr is "un-visited") then
                            set this neighbor as "visited";
                            new-candidates-list=new-candidates-list $\cup$ {nbr} ;
                            set predecessor of nbr to element;
                        end
                        if (nbr == 0) then
                            route-found=true;
                        end
                    end
                end
            end
            if (route-found) then
                $r$=backtrack($s$,$g$);
                Insert-route($r$,$g$,$s$);
            end
            $r$=$r$+1;
        end
        select-route($j$, routes[g][j]);
        Label all nodes in $G$ as "un-visited";
        $N_r^s$={$\phi$};
        num-sources++;
    end
    num-sources=0;
end
function select-route($s$, routes[$g$][$s$]);
begin
select $r_s$= arg $\min_{r \in routes[g][s]}(\max_{k \in N_r^s}(P_k^t))$;
if (num-sources > 1) then
    aggregator[$g$][$s$]=compare-routes($r_s$,routes[$g$][],num-sources);
end
Compute power consumption $m_r$=$m_s^g * w$ /*$w$ is as defined in ILP*/;
$N_r^s$= get-route-nodes($s$,$r_s$,routes[][]);
Update-Power($N_r^s$,$m_r$) /*including relay power*/;
end;
```

Figure 5.8 A high level description of the CBAH heuristic for RSP2 problem.

adjacency matrix T_R of the virtual graph G. In the adjacency matrix T_R, the entry (i,j) is 1 if there is a link from LA node i to its adjacent LA node j; otherwise it is 0. Let $P_i^t = 0$ be the initial total power consumption at each LA node i in graph G, and let e_{ij} be the weight of the edge (i, j) which is the amount of power consumption required to connect LA nodes i and j as defined in Table 5.3. The details of CBAH operation are discussed next. CBAH will cycle through each group and for each source in a group, it finds a route to BS (node 0) such that power consumption at the network is balanced, and hence the network lifetime is prolonged.

For each source, CBAH inspects its neighboring LA nodes. The set of neighbors are stored in a list called the *candidate-list*, which will be expanded during the route search process. Each member of this list is used to initiate the following hop during the route search process. Each time, one candidate is pulled from the list and its four neighbors are examined for possible expansion of the route search. Whenever a node is inserted in the list, CBAH keeps a pointer to the predecessor LA node, which is required to backtrack the route(s) to the source node in case this node falls on the selected route. CBAH keeps a record of the cumulative power consumption at each LA node i in a variable called P_i^t and the value of P_i^t is updated whenever node i is selected as a member of a route for a source node. The importance of the *candidate-list* is that it allows one to quickly select the next LA node needed to expand the route search for a certain source node. In each expansion step, the contents of candidate-list are refreshed by using an auxiliary list called *new-candidate-list*, which contains the nodes that will be used for the next step in route search.

For each source node s, the route discovery phase of CBAH may find multiple disjoint paths by marking nodes which have been *visited*; thus forbidding the visited nodes from being part of more than one route for the same source node. An extension to this is to allow visited nodes to be inspected, but enforce the rest of the route to be the route from that visited node to the BS. In other words, a visited node can still be visited from nodes that are only labeled as un-visited. This allows the checking for more routes. When the destination node (node 0) is reached, the function *backtrack(s,g)* traverses each discovered route back to the source node by using the pointers set up earlier. The set of discovered routes for each source node is stored in a list called (routes[g][s]) using a function called $Insert\text{-}route(r$, routes[g][s]). The list $routes[g][s]$ is 3-dimensional list that stores groups, sources of each group, and the discovered routes for each source node.

After finding the set of routes for each source s, CBAH calls a function called select-route(j, routes[g][s]) to determine which route among the set of discovered routes (routes[g][s]) source s will be using. After deciding which route source s will use, CBAH checks the allocated routes of other sources of the same group and enforce aggregation at some common intermediate LA nodes with these routes for the sake of minimizing the power consumption. Those common nodes will be acting as aggregator nodes for that group. CBAH saves the set of these aggregation points in the list ($aggregator[g][s]$).

The function (*compare-routes*()) detects the common nodes between the routes of different source nodes that belong to the same group by comparing routes of the current source node with previously stored routes of source nodes of the same group, which have been already allocated routes. Note that once being aggregated, the data stream follows the same path, which is the longest path among the rest of the paths found so far. We also tried using the shortest path, but it seems that longest route strategy performs better. If another source node of the same group joins this path at a later point, the process repeats, i.e., further aggregation is performed at this aggregator node, and the new data aggregate is sent along one route to the BS. This means that once data of multiple streams is aggregated at a certain node, no splitting of the aggregate is allowed. Therefore, upon detecting a common node, CBAH will fuse routes of the source nodes, sharing this node, at this common node and then unify rest of the route. This last process is executed in the function *compare-routes*().

Note that an aggregator node may receive two or more different data streams coming from the same source node that belongs to more than one group, i.e., falls in an overlapping region. Hence, CBAH holds a registry for source nodes that exist in multiple groups in order to distinguish between that data sent by this LA source node for each group it belongs to. Once a route r is selected for a source node s, the function *Update-Power(s,N(r))* which is called from within the function *select-route()* will update the amount of total power consumption at each node i in the route r, P_i^t, $i \in N(r)$ by adding the link cost $e(i,j)$ to the current value of P_i^t. Note that each node, which acts as an aggregator, consumes additional power when relaying aggregated data for other nodes. CBAH updates those nodes as well by inspecting the list (*aggregator*[g][s]). After the routes and the set of aggregation points for each source in a certain group is performed, all LA nodes must be cleaned by resetting the label of each node to *un-visited* before another group is served. At the end of CBAH execution, a set of routes as well as a set of aggregator nodes are determined for each source node in each group of the graph G. The algorithm CBAH takes a time complexity on the order O($\Theta n R_{max}$) to finish in the worst case and a space complexity on the order of $O(n|S_g|)$.

5.6 Energy-Delay Tradeoffs Analysis

Although data aggregation results in fewer data transmissions, there is a tradeoff between energy savings and the delay due to the aggregation process. This potential delay may occur because data from closer sources may have to be buffered at an intermediate MA node in order to be aggregated with data coming from sources that are farther away (multiple data will not be received at the same time). In general, there is a time frame during which data needs to reach the BS. Out-of-date data is of no use, for example, if a tracked object is no longer located in the vicinity when the data is received. However, intermediate nodes do not necessarily delay received data for the same period of time. An intermediate node that receives a sufficient amount of data for aggregation does not need to delay the received data any further. In addition, an intermediate node that is not an aggregation point does not need to delay the data at all. The amount of delay should be selected based on the application or other system factors, e.g., the aggregation function performed on the data. Recall that we are assuming periodic sensing with the same period for all sensor nodes.

In our scheme, the aggregation delay occurs at two levels, local and global. In the local aggregation process, delay can be considered negligible since source nodes are in the same zone and they are able to communicate with their peer LA nodes directly. Hence, the aggregation delay is mainly due to the global data processing at further aggregation points. To find the total delay, however, the aggregation delay must be added to the total processing and communication delays required to reach the BS from that MA node. Nevertheless, we are only interested in finding the aggregation delay, that is, the delay incurred by reporting data from different LA source nodes located at different distances to a certain aggregation node. Note that processing delays at aggregation points will be small when compared to the delay incurred in communicating data to the BS. We will now analyze the aggregation delays associated with the two- and multi-level aggregation schemes presented in the previous sections. We start with the two-level aggregation scheme first. Assume that the ILP or any of the approximation algorithms for the two level scheme have been executed and the set of MAs, $\mathcal{M}$, have been allocated in the two-level aggregation scheme and the set of routes and the set of aggregation points for each group in the multi-level aggregation scheme have been determined. Assume that S^j is the set of LA source nodes associated with the jth MA node, $j = 1, 2, \ldots, p$. Let d_{ij} be the distance (in terms of the number of hops) of the path going from the LA source node $i \in S^j$ to its assigned jth MA node in the graph G. Let D^j_{max} and D^j_{min} be the maximum and the minimum distance over all values of d_{ij} with respect to jth MA node, respectively, i.e.,

$$D^j_{max} = \max_{i \in S^j} d_{ij} \quad , \quad D^j_{min} = \min_{i \in S^j} d_{ij}$$

If the aggregation function is duplicate suppression, the aggregation delay incurred, T_{ds}, will be simply the maximum time needed to receive the first unique packet at an MA node calculated over the set of all MA nodes. Each node only passes the first unique packet and suppresses subsequent packets with identical sequence numbers. Hence,

$$T_{ds} = \max_{j \in \mathcal{M}} [\min_{i \in S^j} d_{ij}]$$

In the case of general aggregation functions (e.g. maximum, minimum, average), the aggregation delay incurred, T_{gs}, will be the maximum of the difference in number of hops between an LA source node and its assigned MA node evaluated over all pairs of LA source-MA nodes. Hence,

$$T_{gs} = \max_{j \in \mathcal{M}} [D^j_{max} - D^j_{min}]$$

In the multi-level aggregation scheme, the latency will be proportional to the number of hops between the data aggregation point from the farthest LA source node reporting data to the last aggregation node in the aggregation tree for each group taken over the set of all groups. Let A_f^g be the final aggregation point for all traffic coming from group $g \in \Theta$. Let (t^g) be the maximum time required to reach node A_f^g from all LA sources in group g, and define t_{max} as the maximum delay time due to aggregation taken over all groups in the graph G. Let $D^g(i)$ be the delay time taken by data coming from ith LA source node of group g to reach node A_f^g. The delay $D^g(i)$ is computed by finding the number of hops taken by LA source node i, $i \in S_g$ to reach the last aggregation point, A_f^g, for the group g. Then, the overall maximum aggregation delay is given by,

$$t_{max} = \max_{g \in \Theta}(t^g)$$

and

$$t^g = \max_{i \in S_g}(D^g(i))$$

5.7 Performance Evaluation

The performance of the algorithms of GRASP were tested with various experimental scenarios. Each experiment corresponds to a random placement of sensors in a fixed network area. We assume a single base-station attempting to gather information from a number of data sources in the network area. The location of the base-station can be arbitrarily chosen. We randomly place sensor nodes in a 50m×50m square field while always insuring that the initial distribution of sensor nodes always results in an initially connected graph, as will be explained below. We also experimented with larger sensor fields to test the performance of various heuristics in large networks. It is assumed that the sensing range is the same as the transmission range which was set to a default value of 20 meters. The sensor field is divided into the appropriate number of zones, which is 30. We consider four scenarios corresponding to the distribution of sensors in the sensing field, which result in z nonempty zones (clusters) that are connected to each other, i.e., the distribution must form a connected virtual graph G. In the four scenarios, z takes values of 6, 8, 10, and 15, respectively. In each nonempty zone, there are on average 10 sensor nodes monitoring the area of that zone. We assume that sensors generate data packets with a packet size of that is exponentially distributed with mean value of 1000 bits.

Each sensor i has a battery with finite, non-replenishable energy, which was set to an initial energy of 2 Joules. Whenever a sensor transmits or receives a data packet, it consumes some energy from its battery. The base station has an unlimited amount of energy. The choice of MAC protocol can completely dominate energy consumptions. We assume that energy-conscious protocols like PAMAS [125] or TDMA-based MAC [102] are used for long-lived sensor networks. Our energy model for the sensors is based on the first order radio models [102, 126], which is described in details here for convenience. In the first order radio model, a fixed amount of energy is spent in transmitting and receiving a packet in the electronics, and an additional amount proportional to the distance between tow nodes is spent in transmitting a packet. The radios can perform power control and hence use the minimum required energy to reach the intended recipients. Due to attenuation with distance, an energy loss model with d_{ij}^2 is used for relatively short distances, where d_{ij} is the distance between sensor nodes i and j. More precisely, a radio dissipates E_{elec}=50 nJ/bit to run the transmitter or receiver circuitry and ε_{amp}=100pJ/bit/m^2 for the transmitter amplifier (Figure 5.9 shows the process of sending k bits packet under the first order radio model). Thus, the energy consumed by a sensor node i in receiving a 1000-bit data packet is 1000*50 nJ/bit= 50μJ, while the energy consumed in transmitting a data packet from sensor i to sensor j is given by $T = 50\mu J + 100nJ/m^2 \times d_{ij}^2$. A link transmission rate of 1 Mbps is assumed. We make the assumption that the radio channel is symmetric such that the energy required to transmit a message from LA node A to LA node B is the same as the energy required to transmit a message from LA node B to LA node A. As for delay on a link, it can be calculated as units of time. On a 1 Mbps link, a 1000 bit message can be transmitted in 1ms. We assume that each unit of delay corresponds to 1ms time. Hence, the delay is 1 unit for each 1000 bit message transmitted. In all experiments, we placed the base-station at the middle and to the right side of the virtual grid, such that BS node is connected to the middle four LA nodes in the right side of G.

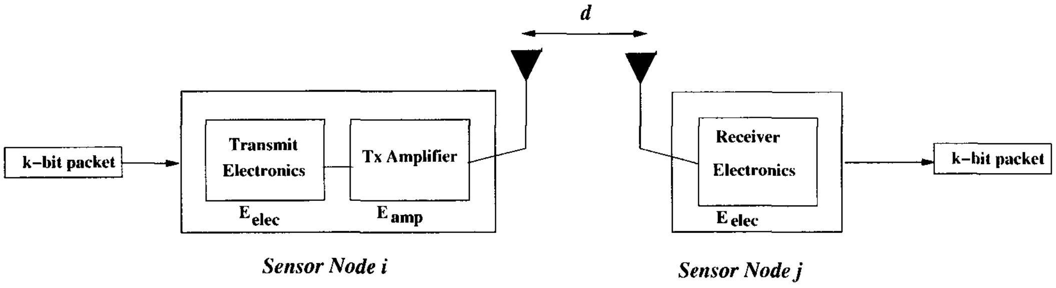


Figure 5.9 First order radio model.

We ignore edge effects where smaller zones on the boundary of the sensor field may exist. However, this simplification is not critical and will not cause any considerable effect on our results. For each data aggregation scheme, the resulting virtual topology (the set of LAs) is then fed into the ILP and approximate algorithms. In both schemes, the algorithms will find the set of routes and MAs for the two-level (2L) aggregation scheme as well as the set of routes and aggregation points for the multi-level (ML) aggregation scheme as described earlier. The ILP problem is solved using the CPLEX linear programming package [51]. In the real problem, the ILP and the heuristics can be solved at the BS node. The set of routes and the aggregation points obtained for both schemes are used for further simulation experiments in order to evaluate the energy-delay tradeoffs as will be explained later in this section. We performed separate sets of experiments to investigate the impact of different parameters (all reported results are averaged over 10 runs). In particular, we studied the following performance issues:

Aggregation versus No-Aggregation: Data aggregation saves energy if the energy required to perform aggregation is lower than the energy required to send raw data (without aggregation) to the BS individually by each source node. We consider the lifetime of the network without aggregation to be the baseline network lifetime, which is taken as 1. We also define the performance metric L as the ratio of the system lifetime achieved using aggregation to that obtained without using aggregation. We refer to L as the lifetime extension ratio. Hence, aggregation saves energy if $L > 1$. We performed separate sets of experiments for both 2L and ML aggregation schemes. The results are shown in Table 5.5 for the 2L scheme for different values of n and number of MAs M, and in Table 5.6 for the ML scheme for different values of n and number of groups Θ.

Table 5.5 Lifetime extension ratio (L) for different 2L aggregation approaches, and for different values of n and M, $\Theta = 3$.

n	M	No Aggregation	GA	LBA	*ILP*
5	3	1	2.48	2.85	3.00
10	5	1	3.99	4.35	5.00
15	7	1	3.77	4.86	5.68
20	9	1	4.26	4.32	4.96

Table 5.6 Lifetime extension ratio (L) for different values of n and Θ in ML schemes.

n	Θ	No Aggregation (NAG)	GA	CBAH	*ILP*
6	2	1	5.48	6.35	9.00
	3	1	17.48	16.53	21.00
8	3	1	11.99	12.05	14.02
	4	1	10.67	11.15	12.00
10	3	1	12.37	12.23	15.31
	5	1	17.72	17.98	20.00
15	5	1	10.26	11.65	14.96
	6	1	21.26	22.17	24.60

As shown in Table 5.5, all schemes with aggregation result in prolonging the lifetime of the sensor network. The value of lifetime extension ratio (L) is the highest with the optimal approach, which can be as large as 5, and sometimes even larger. Out of the four approximate approaches, the LBA approach has the best results. However, the simple greedy approach is not very far behind, which makes it a good candidate for use. Table 5.6 shows values for L for different values of LA nodes, n,

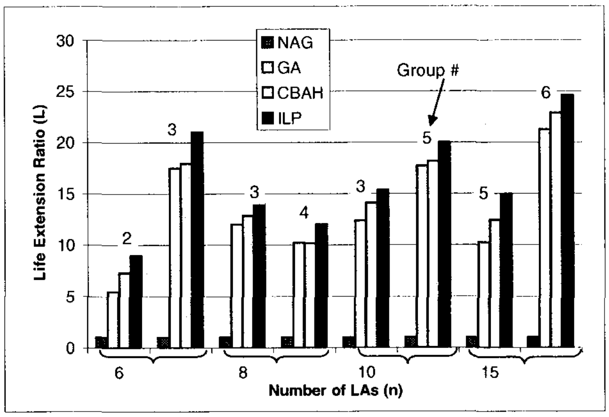


Figure 5.10 Lifetime extension ratio (L) for ML aggregation schemes.

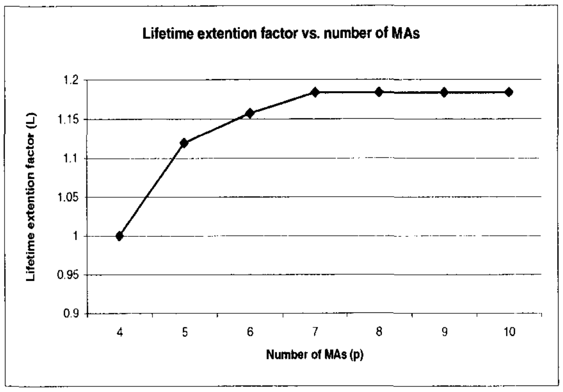


Figure 5.11 The effect of increasing the number of MAs on the extension of the network lifetime when n=15.

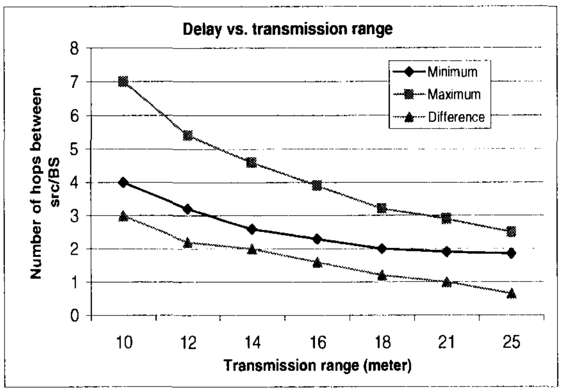


Figure 5.12 The aggregation delay between versus transmission radius in 2L aggregation scheme (LBA).

and when the number of groups, Θ, is varied for every value of n with multiple levels of aggregation allowed. The value of L can be as large as 24 using the ILP especially with a greater number of groups, and sometimes even larger. As noted, the value of L for 2L scheme is lower than the ML scheme. This indicates that further levels of data aggregation can result in greater levels of power savings. To shed light on the performance trend of various algorithms, we plot the contents of Table 5.6 as a graph in Figure 5.10 (the numbers above the bars represent the group numbers). As shown in the figure, the performance of CBAH and GA are not far away from the optimal performance obtained when the ILP is used. However, CBAH run much faster than the ILP especially for large values of n. Table 5.7 and Table 5.8 shows the computation time taken in both two- and multi-level aggregation schemes and for various algorithms. Note that the ILP cannot handle large networks with feasible time while heuristics were able to find solutions quickly.

Table 5.7 Computation time taken by different algorithms in 2L aggregation schemes measured in seconds (Θ=3).

n	M	GA	LBA	ILP
5	3	0.06	0.05	2.74
	4	0.06	0.05	3.52
10	5	0.19	0.14	19.17
	6	0.23	0.16	46.57
15	4	0.53	0.21	100.52
	5	0.61	0.25	213.62
	6	0.69	0.26	499.94
20	8	0.84	0.37	141221.12
	10	0.89	0.40	151232.10
100	20	2.44	2.36	NA

Table 5.8 Computation time taken by different algorithms in the ML aggregation schemes measured in seconds.

n	Θ	GA	CBAH	ILP
6	2	0.48	0.35	8.90
	3	0.56	0.51	21.00
8	3	1.19	1.10	9868.67
	4	1.67	1.15	17821.20
10	3	1.73	1.23	220960.85
	5	1.98	1.81	373219.14
15	5	2.56	2.25	1101682.2
	6	2.98	2.45	1305633.2
50	20	3.96	3.55	NA

Effect of number of MAs: we study the effect of varying the number of MAs, M, for a fixed value of n on the increase in the network lifetime in the 2L aggregation scheme. For this purpose, we fixed n to 15 nodes so that most of the network field is covered and connected. Then, we varied the values of M from 3 to 10 and measured the value of L. Figure 5.11 shows the lifetime extension factor (L) versus the number of MAs. As shown in the figure, when the number of MAs increases, the lifetime is increased until $p = 7$ where no further improvement in the lifetime is obtained, i.e., p will converge to an optimum value as M increases and it levels off after that point ($p = 7$ in this case).

Energy-Delay Tradeoffs Results: In this part, we measure the aggregation delay incurred due to the aggregation process for various schemes. One simple way to consider the aggregation delay is to find

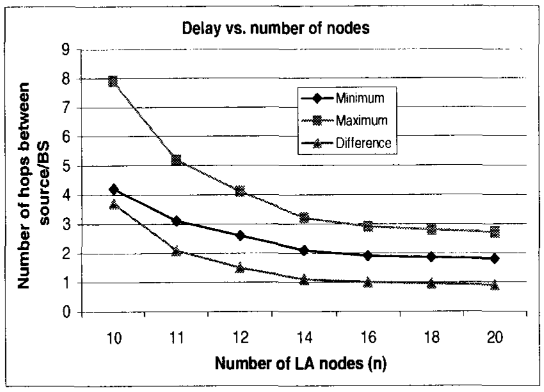


Figure 5.13 Aggregation delay incurred between source(s)/BS pair versus number of LA nodes under the 2L scheme (LBA).

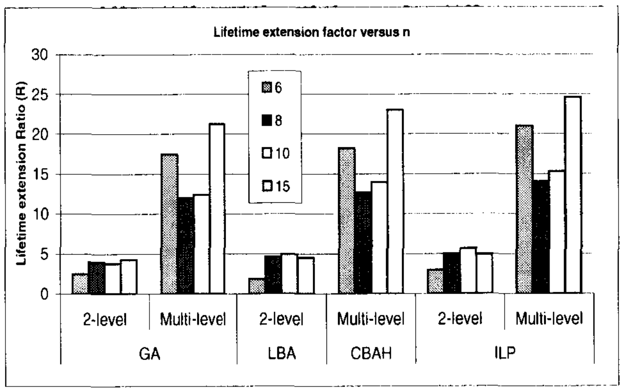


Figure 5.14 The network lifetime extension ratio (L) as the number of LAs increases in the different schemes.

the difference between the minimum and the maximum number of hops traversed by data coming from multiple LA source nodes of the same group and going to the last aggregation node before proceeding to BS taken over all possible groups in the network. The membership of each group is generated randomly and based on geographic proximity such that resulting groups are of comparable sizes. In the 2L aggregation scheme, we studied the difference in hops while varying two parameters: the node transmission range and the number of sensor nodes in the field. For the former, we set $n = 20$ and $M = 9$ and varied the transmission radius from 10 to 25 meters in steps of 5 meters. The number of zones varies according to the transmission range. For the latter case, we set $r = 20$ meters and varied n from 5 to 20 in steps of 5 and M is ranged from 4 to 10 in steps of 2, correspondingly. The number of zones is 30. Figure 5.12 and Figure 5.13 show the minimum, the maximum, and the difference of number of hops traversed by data to reach the MA for each case. As the transmission radius increases, the hop difference decreases and this indicates that the aggregation delay becomes smaller and smaller in the graph G. The increase in the node density helps minimize the difference between the maximum and minimum number of hops. However, as Figure 5.13 shows, after a certain number of sensor nodes are deployed, the nodes are able to maintain almost the same average difference between the maximum and minimum number of hops for all source-MA pairs.

In the ML scheme, we varied the number of LA nodes (n) in the field such that n takes the values 6, 8, 10, 15, respectively in a square area of 50×50 m^2 with r equals to 20 meters and the number of groups Θ equals to 3. All other settings are the same. We found that although ML schemes result in substantial energy savings than the 2L schemes, there is a tradeoff with the aggregation delay.

Figure 5.14 shows the energy savings (lifetime extension ratio (L)) with increasing number of LA nodes for the different schemes under 2L and ML schemes. In all cases, ML schemes achieves multi-fold lifetime extension than the 2L schemes. However, these energy savings come at the expense of more aggregation delays. Figure 5.15 shows the aggregation delays as the number of LA nodes increases in these schemes. The increase in the node density helps to fill the zones in the virtual architecture and increases the node density, and hence the connectivity of the virtual graph. Therefore, the aggregation delay decreases for the 2L aggregation as the number of hops to reach the second-level aggregation point decreases. However, the case is not true for ML aggregation, where many source nodes of the same group may share several aggregation points along the route to BS, and hence more aggregation delays can be experienced. It can be concluded from this comparison that if the sensor network is designed for time critical applications, two-level aggregation scheme would perform better on the expense of additional power consumption. If data gathering and reporting delay is not a concern, multi-level aggregation schemes would be a good choice in this case as these schemes consume less power and hence allow for longer network lifetime.

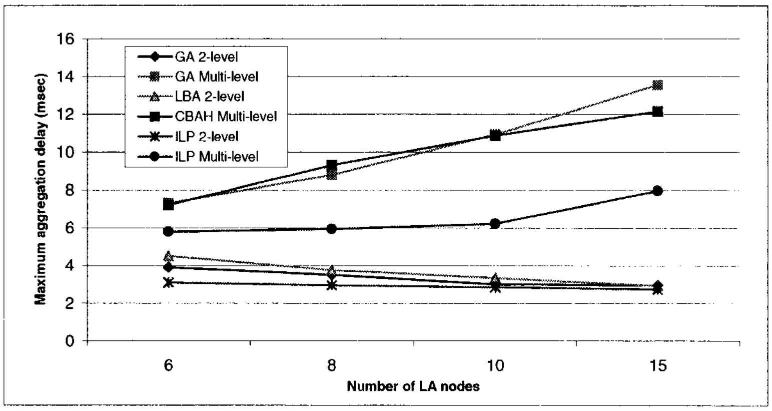


Figure 5.15 Aggregation delays when two- and multi-level data aggregation are used in small networks.

Table 5.9 0.95 Confidence interval for Figures 5.15 and 5.20 with respect to aggregation delays.

n	GA	CBAH	*ILP*
8	9.5±0.13	9.2±0.12	3.1
10	10.1±0.08	10.2±0.03	3.5
100	17.3±0.11	17.5±0.10	-
200	28.3±0.53	31.2±0.26	-
300	41.3±0.15	39.8±0.14	-

Comparison with other data aggregation schemes: In this part, we compare our two-level and

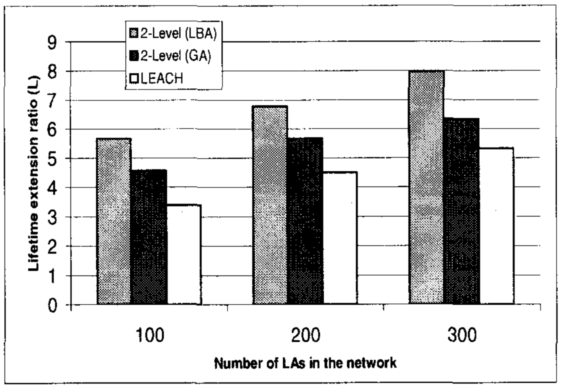


Figure 5.16 Lifetime extension ratio (L) in our two-level aggregation schemes and LEACH.

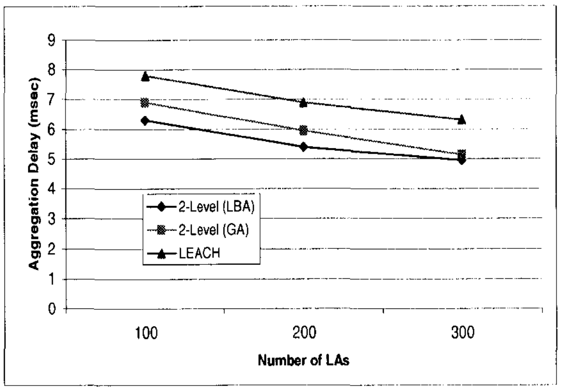


Figure 5.17 Aggregation delays in our two-level aggregation schemes and LEACH protocol.

multi-level routing with data aggregation schemes to some related work in the literature. In particular, we compare our two-level aggregation scheme with LEACH protocol [102] and the multi-level aggregation scheme with the Directed Diffusion paradigm [97]. In the former case, we applied the same simulation scenario of our two-level aggregation scheme in order to compare its performance with LEACH. Recall that LEACH performs aggregation only at the clusterhead nodes and no further aggregation is performed after that. However, in our two-level aggregation scheme, we perform a second level of aggregation at a subset of LAs called Master Aggregators (MAs) in addition to the first data aggregation performed at the clusterheads level. Since the number of dynamic clusters in LEACH is far less than the number of fixed clusters in our clustering approach, we therefore redefine the network lifetime in our scheme as the time until the fist group of LAs is out of energy. This is justified as follows. Since the number of MAs in our scheme is comparable to the number of clusterheads in LEACH, our definition of network lifetime has to be revised as above for a fair comparison with LEACH. Note also that a node may need multiple hops to reach the base-station since clustering is not based on single hop connectivity with clusterhead as the case in our clustering scheme. Unlike our clustering scheme, clusterheads in LEACH are chosen randomly, and without consideration to the sensor node status.

For the multi-level aggregation scheme, we applied the same scenario used in our ML aggregation scheme using the Directed Diffusion (DD) paradigm as per [97]. Our objective in both cases is to study the impact of different numbers of sources and different numbers of groups on the network lifetime and the amount of aggregation delays experienced in the network. In all simulations, we have uniformly distributed 100, 200, and 300 sensor nodes in a 200×200m^2 fixed sensor field. For comparison with LEACH, we set transmission range to 87 meters as per [102]. The BS location is as explained before. In LEACH, we also selected 5% of the nodes to act as LAs as suggested in [102]. Thus, we have 5, 10, and 15 clusters for the 100, 200, and 300 sensor nodes, respectively, while the number of zones (clusters) in

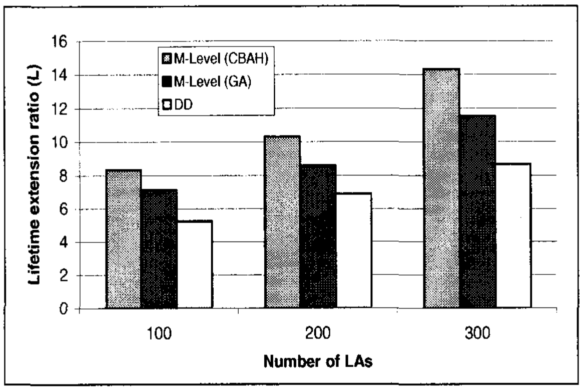


Figure 5.18 Lifetime extension ratio (L) for our multi-level data aggregation schemes and Directed Diffusion (DD). M-Level: Multi-Level.

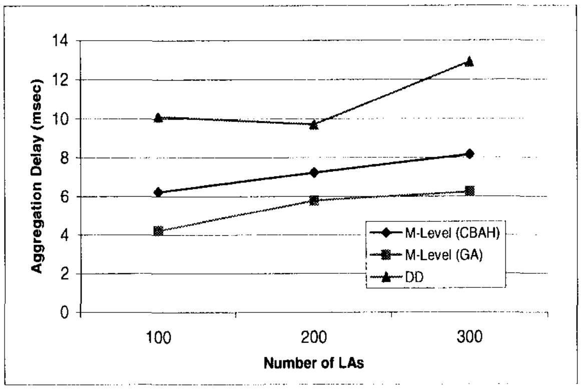


Figure 5.19 Aggregation delays in both our multi-level data aggregation schemes and the Directed Diffusion scheme.

our scheme is fixed and is equal to 25 regardless of the number of nodes used. The simulation results are shown in Figures 5.16 and 5.17. We first look at the lifetime extension ratio (L), shown in Figure 5.16, in our scheme and the LEACH protocol. Our scheme is able to prolong the network lifetime more than LEACH protocol by a factor of more than 1.7 in all simulation scenarios. Furthermore, the aggregation delays associated with our scheme is up to 25% less than that of LEACH as shown in Figure 5.17. Note that in LEACH, clusters are formed dynamically in each round and clusterheads gather data locally and then transmit to the BS possibly in multihops with no further aggregation.

The fixed virtual topology scheme presented in this paper improves on LEACH by saving energy and delay in several stages. At the first level, sensor nodes are transmitting at shorter distances to their LAs compared to nodes transmitting to a possibly faraway clusterhead in the LEACH protocol. For example when the number of nodes is 100, only 5 clusters of nodes are formed or 5 clusterheads are elected in LEACH. In LEACH, a clusterhead needs to wait for all nodes to transmit their data to it before sending the aggregate form to the BS. Hence, the aggregation delay for all nodes to transmit to the cluster-head is the maximum number of nodes in any of the 5 clusters.

Next, we focus on the energy-delay tradeoffs in the multi-level data aggregation schemes compared to Directed Diffusion (DD). Figures 5.18 and 5.19 plot the lifetime extension ratio (L) and aggregation delays for our approximate schemes and the DD paradigm, respectively, as the number of LA source nodes increases. For a fixed number of LA source nodes and a fixed number of groups, our schemes consume less energy than DD, and hence extends the network lifetime. This is due to two reasons. First, query flooding in our scheme is confined to only horizontal and vertical directions, while in DD a query propagates throughout the whole network field before some paths are re-enforced. Second, our scheme searches for routes that balance the power consumption in the network, while DD makes no distinction between routes used that will carry data to the BS. Figure 5.19 plots the data aggregation delay experienced in our schemes and in the DD when the number of LAs increases. As shown in the figure, our schemes experience lower aggregation delays than DD as the number of LAs increases. This is because in DD data forwarding paths from different sources may cross or overlap with each other anywhere in the network area, thus there are more interferences when the number of sources is large. Whereas in our scheme each LA source node sends data on the virtual grid, thus data flows on the grid faster and the routes are selected to balance power consumption in the network.

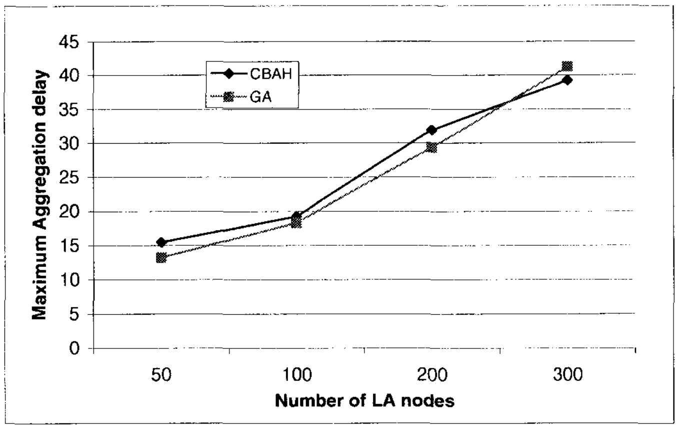


Figure 5.20 Aggregation delays for multilevel aggregation schemes in large networks.

Data Aggregation/Routing in Large WSNs: In this part, we demonstrate the effectiveness of the approximate algorithms in large WSNs in terms of the area of the sensor field and large number of sensor nodes. Recall that the ILPs can only solve the RSP1 and RSP2 problems in small to medium sized networks with feasible time. Hence, for large WSNs, we only focus on the approximate algorithms and using the multilevel aggregation schemes. We generated graphs of 100, 200, and 500 LA nodes in an area of 500×500 m^2. All other settings are the same. We then varied the number of groups in each graph and measured the lifetime extension ratio L in both approximate schemes. Again, L is set to 1 when no aggregation is performed. Table 5.10 shows the results. Both algorithms are able to achieve significant energy savings that prolong the network lifetime. CBAH performs slightly better than GA since the chromosome size in GA becomes large as n increases, and hence GA needs significant computation time to converge to the best individual. In experimenting with the two approximate algorithms, we noticed that CBAH produces good results in shorter time than GA. However, GA can produce better results at the expense of much longer run times.

We also measured the aggregation delay incurred in both algorithms. Figure 5.20 shows the aggregation delay using multi-level data aggregation for the case when n takes the values 50, 100, 200, and 300 and when Θ=30. Group membership is based on geographic proximity. As shown in the Figure,

Table 5.10 Lifetime extension ratio (L) for approximate algorithms in large WSNs and for different values of n and Θ.

n	Θ	No Aggregation	GA	CBAH
100	10	1	25.18	26.25
	20	1	31.42	31.95
	30	1	37.28	39.45
200	30	1	25.79	27.63
	50	1	37.61	41.26
300	40	1	42.34	45.21

the aggregation delay increases in both algorithms with some minor differences as the number of LAs increases. This is due to the increase in the number of data aggregation levels, which result in that data belonging to the same group is delayed at multiple LA nodes in the way to the BS. However, as the number of LA nodes in the network increases, the relative energy savings also increases (see Table 5.10), which again illustrates the trend of achieving greater energy savings at the expense of more aggregation delays.

5.8 Chapter Summary

In this chapter, we studied the maximum lifetime data gathering problem in WSNs. We showed that cluster-based algorithms along with data aggregation and in-network processing can achieve significant energy savings in WSNs. This has a direct effect on prolonging the network lifetime. We developed GRASP (Grid-based Routing and Aggregator Selection Protocols), a scheme for WSNs that combines the ideas of fixed cluster-based routing together with application-specific data aggregation in order to enhance the wireless sensor network performance in terms of extending the network lifetime, while incurring acceptable levels of latency in data aggregation. Within GRASP, we have presented exact as well as approximate algorithms that solve the joint problem of optimal routing with data aggregation for the sake of maximizing the network lifetime. Our results show that, when compared to other approaches in the literature, the proposed scheme is able to improve the network lifetime while incurring acceptable levels of latency in data aggregation, and without sacrificing quality. Hence, GRASP can attain the energy and latency efficiency needed for wireless sensor networks.

CHAPTER 6 Conclusions and Future Work

Due to advances in wireless technology and energy-efficient design, the widespread use of mobile and handheld devices is likely to popularize ad hoc networks, or so called Infrastructerless Wireless Networks (IWNs). While these networks allow mobility and flexibility, they add constraints that are not found in a wired environment. Among the classes of IWNs, Mobile Ad hoc Networks (MANETs) and Wireless Sensor Networks (WSNs) are currently receiving extensive research in order to achieve the ubiquitous computing paradigm perceived from such networks. In this dissertation, we designed and evaluated various cluster-based protocols that can support Quality of Service (QoS) provisioning in MANETs and energy-efficient communication in WSNs.

6.1 Summary of Contributions

The work described in this dissertation demonstrated the advantages of cluster-based protocols and architectures by designing and evaluating protocols for two different IWNs classes: Mobile Ad hoc Networks (MANETs) and Wireless Sensor Networks (WSNs). In particular, we addressed cluster-based QoS routing in MANETs and energy-efficient routing in WSNs.

In Chapter 2, we investigated providing a stable topology with low overhead. We presented a novel clustering scheme that results in a rectilinear virtual topology, called Virtual Grid Architecture (VGA). Members of VGA, namely, the clusterheads (CHs) are elected using a CH election algorithm with various strategies. The clustering approach is simple, modular, scalable, stable, and close to optimal. Different routing schemes for MANETs can be implemented over VGA. We showed how this stable routing architecture, namely, VGA can be utilized to help in performing simple and efficient routing in MANETs using two routing techniques, namely, on-demand and transitive closure based routing techniques. We also presented a scheme that enforces or stimulates node cooperation in the CH election process and in the routing process over VGA. Although we selected the zones comprising the network area to be square which results in a simple and regular virtual topology, other regular shapes can also be used (e.g. hexagons, equilateral triangles). Each regular shape has its own merits and limitations.

In Chapter 3, we conducted a comprehensive performance study of the VGA clustering approach. In particular, we studied the tradeoffs between the simple VGA clustering and an optimal clustering in both homogeneous and heterogeneous MANETs. For homogeneous MANETs and under the assumption of large number of nodes, our topology control scheme, VGA, is proven to be scalable, robust, and energy-efficient. In fact, our results show that, albeit being simple, the performance of VGA is close to optimal. Since stability and connectivity are two important properties of any protocol designed for MANETs, we also analyzed the connectivity and stability of VGA. Our results showed that VGA is stable architecture and it is robust against node dynamics. We showed that the probability of a CH being disconnected in VGA is low.

In Chapter 4, we presented two QoS routing protocols for MANETs. The first protocol, called Virtual Grid Architecture Protocol (VGAP), is a QoS routing scheme for heterogeneous MANETs that works on top of VGA. VGAP utilizes the interaction of the bottom three layers of protocol stack in MANETs in order to enhance the network performance and support statistical quality of service guarantees in terms of bandwidth and end-to-end delay guarantees. The simulation results of VGAP showed improved performance in terms of the number of admitted calls and packet delivery ratio. Moreover, the network showed more resilience to link failure and topology variations, which is a result of VGA stability. The second QoS routing protocol, called Virtual Routing with Feedback (VRF), addresses load balancing among the routes in the network while at the same time provides QoS guarantees. VRF utilizes a feedback mechanism to discover routes with less end-to-end delays or routes with higher energy reserve. The feedback mechanism of VRF is found to have profound effects on the packet delivery ratio and the call success ratios in MANETs that employ VGA clustering approach.

In Chapter 5, we introduced and analyzed an approach to prolong the lifetime of wireless sensor networks (WSNs) using the VGA of Chapter 2 for routing purposes. The scheme, called Grid-based Routing and Aggregator Selection Protocol (GRASP), incorporates data-centric routing and application-specific processing inside the network (e.g., data aggregation) to extend the network lifetime by minimizing the number of transmissions. Given that, in wireless sensor networks, the communication cost in terms of power consumption is several orders of magnitude higher than the computation cost [127], GRASP can achieve significant energy savings with in-network data aggregation and in-network processing. The benefit of data aggregation in GRASP has been confirmed by using algorithmic approach and with extensive experimentation. It has been demonstrated that GRASP is able to achieve low energy dissipation and low latency without sacrificing data quality in WSNs. In GRASP, data aggregation is performed at two- and multiple-levels using the fixed virtual backbone. Exact as well as several approximate algorithms for the joint problem of data routing and aggregation points selection were proposed. Besides being very fast and scalable to large sensor networks, the approximate algorithms produced results which are not far from the optimal solution. Numerical results in both schemes showed that substantial energy savings can be obtained while incurring low aggregation delays when compared to related work in the literature.

6.2 Future Directions

There is still much work to be done in the area of protocols for Infrastructerless Wireless Networks (IWNs). Despite various advantages and unlimited application chances, MANETs are still far from being deployed on large scale commercial basis. This is because of some fundamental problems that remain unsolved or need better solutions. The protocols and techniques presented in this dissertation are attempts to enhance the performance of IWNs. Research on improving data transmission over MANETs with QoS guarantees and report data with less power consumptions in WSNs is still needed. However, the design space of VGA is so large and non-trivial that we have only explored some of the design issues in this work. In the following, we suggest some of the most important issues that can be further investigated in the area of QoS provisioning in MANETs and energy-efficient routing in WSNs.

- **Mobile Nodes Position Identification:** The provisioning of QoS in VGAP and VRF is dependent on the position of mobile nodes, which were used to create VGA. Each mobile node must know its adjacent neighbors and other mobile nodes. This has a direct impact on the QoS routing protocol. The use of external entities (e.g., GPS) is promising despite the need to solve the problems associated with the use of GPS system. A suitable position determination system, and position upgrade mechanism is still required. Although we proposed an enhanced mobility model in Chapter 2 of this dissertation, there is a need for a mobility testbed that can be used to test for real life mobility patterns.

- **Internet-MANET Integration:** The integration between MANET and the existing global information infrastructure, the Internet, is essential. Being able to connect to wireline network, e.g., Internet, can help in the QoS routing when a MANET becomes partitioned, for example. VGAP uses a modified OSPF to perform QoS routing in MANETs. As known, OSPF is used in the Internet, which facilitates the integration between MANETs and the Internet. However, there is a need for specifications in VGAP that would identify how the two networks can interact.

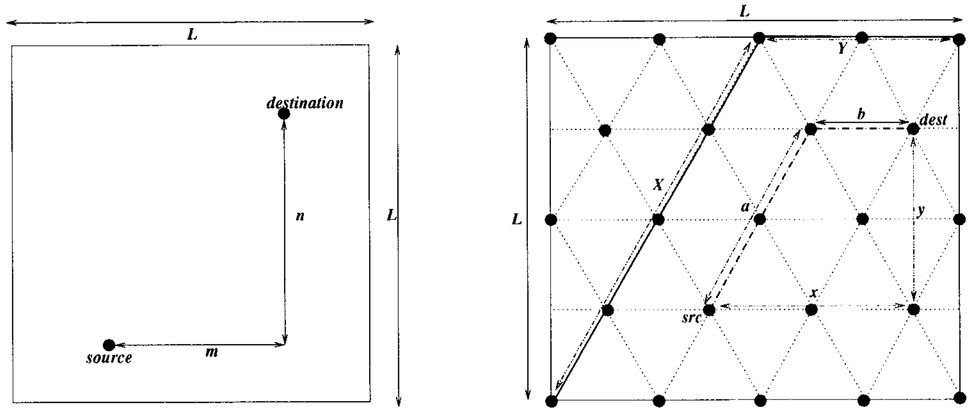
- **Nodes Cooperation:** Some mobile nodes may misbehave by agreeing to forward packets and then failing to do so or by acting selfishly and refusing to be a clusterhead. A node may misbehave because it is overloaded, selfish, malicious, or broken. Node cooperation is essential for the network to perform its duties. The scheme presented in Chapter 2 to enforce nodes cooperation is an attempt in this direction. Although the scheme works efficiently, we think that there should be a form of network-level or link-layer security aspects in the protocol such that malicious retransmissions, manipulation, snooping, and redirection of packets are prevented. This is another thread that we intend to pursue in a future work.

- **Protocols for Sensor Networks:** The protocols developed in this research (Chapter 5) are for scenarios where the sensors have correlated data. However, there are important applications of sensor networks where this is not the case. For example, sensor networks for medical monitoring applications may have different sensors located on and/or in the body to monitor vital signs. These networks will most likely focus on maximizing quality above all parameters and loss of information will not be acceptable. Therefore protocol architectures need to be developed to support the unique considerations of these networks.

  While we discussed the use of data aggregation in saving energy in WSNs, there is a need for better, faster, and more accurate data aggregation modeling and classification. For example, future sensor networks may contain cameras, microphones, and seismic sensors, each obtaining data about events in the environment. It is important to be able to aggregate this data at different levels. Within the GRASP scheme, the following are some possible courses of action:

  - **Exploit spatial diversity and redundancy:** Since sensor nodes are prone to failure, we want our routing techniques to explicitly employ fault tolerance methods in order to extend network lifetime. Moreover, in large WSNs, nodes will span a network area that might be large enough to provide spatial communication between sensor nodes. This also motivates the need to have time and location synchronization among nodes.
  - **Mobility:** Our experiments on WSNs impose the condition of fixed sensor nodes and base-stations. However, the impact of node mobility on GRASP has not been investigated and warrant further research.
  - **Quality of Service:** A plausible next step after the design of the GRASP is to provide quality of service according to the packet importance (priority) or bounding delay on packet delivery. Related to QoS is the congestion control issue. Given bandwidth limitations, a congestion control mechanism is unavoidably necessary in WSNs. A possible study would combine congestion control mechanisms with data aggregation design in GRASP to further enhance the performance of the network while conserving energy at the same time.
  - **Secure Routing:** Without basic security measures in place, the application of sensor networks will remain limited. The aspect of in-network aggregation complicates the design of a secure routing protocol. In-network processing makes end-to-end security mechanisms harder to deploy because intermediate nodes need direct access to the contents of the messages. An in-depth analysis of protocols' security in the context of GRASP could help alleviate the basic security concerns of WNSs users.

APPENDIX A Proof of Proposition 1



(a) VGA routing

(b) optimal rouitng

Figure A.1 Figure used in the proof of Proposition 1

Proposition: If the network area is large and is equal to $L \times L$ square meters, mobile nodes are uniformly distributed in the network area, and each node has a transmission range of r meters, then the worst case path length (PL) of VGA, D-VGA, and Optimal routing (in terms of the number of hops) is given by:

$$PL_{VGA}^{max} = \frac{2L\sqrt{5}}{r} - 1\ ,\ PL_{D-VGA}^{max} = \frac{L\sqrt{8}}{r}\ ,\ PL_{Opt}^{max} = \frac{L}{r}(\frac{1+\sqrt{3}}{\sqrt{3}}),$$

while the average case path length is given by:

$$PL_{VGA}^{avg} = \frac{\sqrt{5}L}{r}\ ,\ PL_{D-VGA}^{avg} = \frac{5\sqrt{2}L}{3r}\ ,\ PL_{Opt}^{avg} = \frac{(3\sqrt{3}+11)L}{18r}.$$

Proof. Divide the network area into $Z = M \times M$ smaller zones. Let the horizontal length of an arbitrary route be m rows (zones) and vertical length be n columns (zones). With VGA routing, the zone side length is $\frac{r}{\sqrt{5}}$, then

$$M_{VGA} = \frac{L\sqrt{5}}{r},$$

and the worst case number of hops over VGA is $2M$-1 which is equal to

$$PL_{VGA}^{max} = \frac{2L\sqrt{5}}{r} - 1$$

This can be $\approx \frac{2L\sqrt{5}}{r}$ for large L. Adding diagonal routing will decrement the zone side length to $\frac{r}{2\sqrt{2}}$, hence for D-VGA,

$$M_{D-VGA} = \frac{L\sqrt{8}}{r},$$

and the worst case number of hops is = max(m,n), and hence

$$PL^{max}_{D-VGA} = \frac{L\sqrt{8}}{r}$$

for large L. Therefore, in the worst case, the rectilinear routing increases the number of hops by a factor of $\frac{\sqrt{20}}{\sqrt{8}} = 1.6$. For the optimal case, when the zones are hexagons, the worst-case path length can also be calculated as follows. The virtual topology connectivity for the optimal case is shown in Figure A.1(b), where each clusterhead can communicate with 6 possible neighbors. The worst case number of hops (PL^{max}_{Opt}) in this case can easily be found by noting that the angles of movement are $0^o, 60^o, 120^o, 180^o, 220^o, and\ \ 300^o$. The worst-case path length (in terms of the number of hops) is $X + Y$ where X and Y are shown in Figure A.1(b) and they can be easily obtained as $X = \frac{2L}{\sqrt{3}}$ and $Y = L(1 - \frac{1}{\sqrt{3}})$, hence the worst-case path length is

$$PL^{max}_{Opt} = \frac{L}{r}(\frac{1+\sqrt{3}}{\sqrt{3}})$$

Now, we will find the average hop length of a route (i.e., route length) between a source-destination (s, d) pair and for both diagonal and rectilinear routing, and compare it to the optimal route length. Let x and y be two random variables representing the length of a route in the horizontal and vertical directions, respectively (see Figure A.1(a)). Let $f_x(x)$ and $f_y(y)$ be the probability distribution functions of x and y, respectively. Further, assume that x and y are uniformly distributed in the network area. Hence, $f_x(x)= f_y(y)= \frac{1}{L}$. Then,

$$\text{VGA routing} \ : \ m = \frac{x}{r/\sqrt{5}}\ , \ \ n = \frac{y}{r/\sqrt{5}} \quad \text{D-VGA routing} \ : \ m = \frac{x}{r/\sqrt{8}}\ , \ \ n = \frac{y}{r/\sqrt{8}}$$

and the number of hops for both cases is:

D-VGA=$max(m, n)$ VGA routing=$m + n - 1$

assuming $m+n >> 1$, and also assuming that the horizontal and vertical granularity is very large, then we can assume m and n are continuous. Therefore, the average path length, APL for the VGA routing is:

$$PL^{avg}_{VGA} = \int_0^L mf_x(x)dx + \int_0^L nf_y(y)dy = \int_0^L \frac{x}{r/\sqrt{5}}\frac{1}{L}dx + \int_0^L \frac{y}{r/\sqrt{5}}\frac{1}{L}dy = \frac{\sqrt{5}L}{r}$$

For D-VGA routing, we have

$$PL^{avg}_{D-VGA} = \int_0^L (\int_{x=0}^y nf_x(x)dx + \int_{x=y}^L mf_x(x)dx)f_y(y)dy$$

$$= \int_0^L (\int_{x=0}^y \frac{y}{r/\sqrt{8}}\frac{1}{L}dx + \int_{x=y}^L \frac{x}{r/\sqrt{8}}\frac{1}{L}dx)\frac{1}{L}dy = \frac{5\sqrt{2}L}{3r}$$

What is left is to find the average path length of the optimal case. Let x and y represent the length of a route in the horizontal and vertical directions, respectively (see Figure A.1(b)). It can be easily seen that the number of hops between an arbitrary source-destination pair is the sum of the two segments

$a + b$, where a and b can easily be found as $a = \frac{2y}{\sqrt{3}r}$. Note that the value of b is dependent on the relation between x and y such that:

$$b = \begin{cases} (x - \frac{y}{\sqrt{3}})/r & , if\ x \leq \frac{y}{\sqrt{3}} \\ (\frac{y}{\sqrt{3}} - x)/r & , if\ x \geq \frac{y}{\sqrt{3}} \end{cases}$$

Hence the average path length of the optimal case, PL_{Opt}^{avg}, can be found by finding:

$$PL_{Opt}^{avg} = \int_0^L \{\frac{2y}{\sqrt{3}r} + \int_{x=0}^{\frac{y}{\sqrt{3}}} (\frac{y}{\sqrt{3}} - x)\frac{1}{r} f_x(x)dx + \int_{x=\frac{y}{\sqrt{3}}}^{L} (x - \frac{y}{\sqrt{3}})\frac{1}{r}\frac{1}{L} f_x(x)\} f_y(y)dy$$

Recall that $f_x(x)$= $f_y(y)$= $\frac{1}{L}$, then

$$PL_{Opt}^{avg} = \int_0^L \{\frac{2y}{\sqrt{3}r} + \int_{x=0}^{\frac{y}{\sqrt{3}}} (\frac{y}{\sqrt{3}} - x)\frac{1}{r}\frac{1}{L}dx + \int_{x=\frac{y}{\sqrt{3}}}^{L} (x - \frac{y}{\sqrt{3}})\frac{1}{r}\frac{1}{L}dx\}\frac{1}{L}dy = \frac{(3\sqrt{3} + 11)L}{18r}$$

□

APPENDIX B Routing Techniques Comparison in WSNs

	Mobility	*Power Requirement*	*Localization*	*Data aggregation*	*Location awareness*	*Length of path*	*Scalability*	*State complexity*	*Computation & communication overhead*	*Classification*
LEACH	Fixed BS	High Power requirement for BS	Yes	Yes	No	N/A	Good	Cluster heads maintains it	Setting up and maintaining Cluster	Clustering
PEGASIS	Fixed BS	High Power requirement for BS	Yes	Yes	No	Fixed	No	Very simple	Same as *LEACH* plus chain setup	Reactive / Clustering
TEEN	Fixed BS	High Power requirement for BS	Yes	Yes	No	N/A	Good	Very simple	Same as *LEACH*	Reactive / Clustering
SPIN	Could support	Limited	No	Could	No	SP	Limited	Very simple	flooding ADV and data	Proactive / flat
Directed Diffusion	Limited	Limited	Yes	Yes	No	Almost SP	Limited	Per pair of BS & SN	Too much overhead	Proactive / flat
Minimum Cost Forwarding	No	Not specified	No	No	No	SP in terms of cost	Good	Per sink	Setting up cost field & one hop broadcasting	Proactive / flat
TTDD	Mobile BS	Limited	Local query flooding	No	Yes	At most $\sqrt{2}$ times of SP	Could	Per event only on dissemination points	Setting up grid & flooding query in local cell & trajectory forwarding	Proactive / clustering
Random Walks	No	Rechargeable	Yes	No	No	Random	Good	Per sink or stateless	Computing lattice coordination	Reactive / flat
Rumor Routing	Very limited	Not specified	Yes	No	No	Random	Good	Per event or query only on random nodes	Setting up event adv path & possible query flooding	Hybrid / flat

Table B.1 A comprehensive comparison between routing protocols in WSNs. BS:Base-Station, SN: Sensor Node, SP: Shortest Path.

BIBLIOGRAPHY

[1] The 3rd Generation Partnership Project (3GPP) Web Site. *http://www.3gpp.org*, (date retrieved: March 6, 2004).

[2] X. Bangnan, S. Hischke, B. Walke, *The role of ad hoc networking in future wireless communications*, International Conference on Communication Technology (ICCT 2003), Vol: 2, Pages: 1353-1358.

[3] M. Abbott and L. Peterson, *Increasing network throughput by integrating protocol layers*, IEEE/ACM Transactions on Networking, Vol. 1, No. 5, Page(s): 600-610, October 1993.

[4] L. Robinson. *Japan's new mobile broadcast company: Multimedia for cars, trains, and hand-helds*, in Advanced Imaging, Page(s): 18-22, July 1998.

[5] J. Jubin and J.D. Tornow. *The DARPA packet radio network protocols*, proceedings of IEEE, Vol. 75, No. 1, Page(s): 21-32, 1987.

[6] D. Beyer, M. Vestrich, and J.J. Garcia-Luna-Aceves. *The rooftop community network: Free, high-speed network access for communities.* proceedings of First 100 Feet Conference, Arlington, Virginia, USA, October 1996.

[7] K. Sairam, N. Gunasekaran, S. Redd, *Bluetooth in wireless communication*, IEEE Communications Magazine, Vol.:40, No.: 6, Pages: 90-96, June 2002

[8] K. Negus, A. Stephens, J. Lansford, *HomeRF: wireless networking for the connected home*, IEEE Personal Communications, Volume: 7, No.: 1, Feb. 2000.

[9] H. Xiaoyan, X. Kaixin, M. Gerla,*Scalable routing protocols for mobile ad hoc networks*, IEEE Network, Volume: 16 No.: 4, July-Aug. 2002 Page(s): 11-21.

[10] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cyirci, *Wireless sensor networks: A survey*, Computer Networks, Vol. 38, No. 4, Page(s): 393-422, 2002.

[11] W. Liao, Y. Tseng, K. Shih, *A TDMA-based bandwidth reservation protocol for QoS routing in a wireless mobile ad hoc network*, ICC 2002. Volume: 5, Page(s): 3186-3190, 2002.

[12] D. Ayyagari, A. Michail, A. Ephremides, *A unified approach to scheduling, access control and routing for ad-hoc wireless networks*, VTC 2000, Volume: 1, Page(s): 380-384, 2000.

[13] V. Kawadia and P. R. Kumar, *Power control and clustering in ad hoc networks.* proceedings of INFOCOM 2003, Volume: 1, Page(s): 459-469, April 2003.

[14] Y. B. Ko and N. Vaidya. *Location-aided routing (LAR) in mobile ad hoc networks.*, Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM), Page(s): 66-75, 1998.

[15] S. Capkun, M. Hamdi, J. Hubaux, *GPS-free positioning in mobile ad-hoc networks*, Proceedings of the 34th Annual Hawaii International Conference on System Sciences, Page(s): 3481-3490, 2001.

[16] J. Li, J. Jannotti, D. D. Couto, D. Karger, and R. Morris. *A scalable location service for geographic ad hoc routing.* In Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM00), Page(s): 120-130, 2000.

[17] S. Chakrabarti and A. Mishra, *QoS issues in ad hoc wireless networks*, IEEE Communications Magazine, 39(2), Page(s): 142-148, February 2001.

[18] M. Mauve, A.Widmer, H. Hartenstein,*A Survey on position-based routing in mobile ad hoc networks*, IEEE Network, Vol. 15, No. 6, Page(s): 30-39, 2001.

[19] T. Camp, J. Boleng, V. Davies,*A survey of mobility models for ad hoc network research*, Wireless Communications & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications, 2002. Page(s): 332-347, 2002.

[20] I. Stojmenovic, *Position-based routing in ad hoc networks*, IEEE Communications Magazine, July 2002, Page(s): 128-134, 2002.

[21] D. J. Baker and A. Ephremides, *The architectural organization of a mobile radio network via a distributed algorithm*, IEEE Transactions on Communications, Vol. 11, Page(s): 16941701, November 1981.

[22] B. Das and V. Bharghavan, *Routing in ad hoc networks using minimum connected dominating sets*, ICC' 97, Montreal, Canada, June 1997.

[23] M. Gerla and J. T.-C. Tsai. *Multicluster, mobile, multimedia radio network*, ACM Baltzer Journal of Wireless Networks, Vol. 1, No. 3, Page(s): 255265, 1995.

[24] C.R. Lin and M. Gerla, *Adaptive clustering for mobile wireless networks*, IEEE J. Selected Areas in Comm., Vol. 15, No. 7, Page(s): 1265-1275, Sept. 1997.

[25] A. Farago, I. Chlamtac, S. Basagni, *Virtual path topology optimization using random graphs*, Proceedings IEEE INFOCOM '99, New York-USA.

[26] Y. Xu, J. Heidemann, and D. Estrin,*Geography-informed energy conservation for ad hoc routing*, International Conference on Mobile Computing and Networking (MOBICOM01), Pages: 70-84, 2001.

[27] S. Banerjee and S. Khuller, *A Clustering scheme for hierarchical control in multi-hop wireless networks*, Proc. IEEE INFOCOM 01, Page(s): 1028-1037, 2001.

[28] T.-C. Hou and T.-J. Tsai, *An access-based clustering protocol for multihop wireless ad hoc networks*, IEEE J. Selected Areas Comm., vol. 19, No. 7, Page(s): 1201-1210, July 2001.

[29] W. Liao, Y. Tseng, J.P. Sheu, *GRID: A fully location-aware routing protocols for mobile ad hoc networks*, Telecommunication Systems, Vol. 18, No. 3, Page(s): 37-60, 2001.

[30] A.D. Amis, R. Prakash, T. H. P. Vuong, and D. T. Huynh. *Max-Min D-cluster formation in wireless ad hoc networks*, proceedings of INFOCOM 2000, Vol. 1, Page(s): 32-41, March 2000.

[31] K.M. Alzoubi, P.-J. Wan and O. Frieder, *New distributed algorithm for connected dominating set in wireless ad hoc networks*, Proceedings of HICSS, 2002.

[32] J. Wu and H. Li, *On calculating connected dominating set for efficient routing in ad hoc wireless networks*, Proc. of the 3rd International Workshop on Discrete Algothrithms and Methods for MOBILE Computing and Communications, 1999, Seattle, WA USA, Page(s): 7-14.

[33] X. Cheng and D. Du, *Virtual backbone-based routing in multihop ad hoc wireless networks*, Technical Report, Department of Computer Science and Engineering, University of Minnesota, Minneapolis, 2002.

[34] A.M. Safwat and H.S. Hassanein, *Infrastructure-based routing in wireless mobile ad hoc networks*, Computer Communications, Vol. 25, No. 3, 2002, Page(s): 210-224.

[35] J. Wu, F. Dai, M. Gao, and I. Stojmenovic *On calculating power-aware connected dominating set for efficient routing in ad hoc wireless networks*, Journal of Communications and Networks , Vol. 5, No. 2, March 2002, Page(s): 169-178.

[36] Y-B Ko and N. H. Vaidya, *Location-Aided Routing (LAR) in mobile ad hoc networks*, In Proceedings of the 4th ACM International Conference on Mobile Computing and Networking, MOBICOM 98, Page(s): 66-75, Oct. 1998.

[37] P. Sinha, R. Sivakumar, and V. Bharghavan, *CEDAR: a core-extraction distributed ad hoc routing algorithm*, Proceedings of INFOCOM 1999.

[38] A.B. McDonald and T.F. Znati, *A Mobility-based framework for adaptive clustering in wireless ad hoc networks,* IEEE J. Selected Areas in Comm., Vol. 17, Page(s): 1466-1487, Aug. 1999.

[39] C. Bettstetter, *On the minimum node degree and connectivity of a wireless multihop network*, Proc. MobiHoc 2002, Page(s): 80-91.

[40] T. K. Philips, S. S. Panwar, and A. N. Tantawi, *Connectivity properties of a packet radio network model,* IEEE Trans. on Info. Theory, Vol. 35, Page(s): 1044-1047, September 1989.

[41] H. C. Yang and M. Alouini, *Closed-form formulas for the outage probability of wireless communication systems with a minimum signal power constraint*, IEEE Trans. on Vehicular Technol. Vol. 51, Page(s): 1689-1698, November 2002.

[42] D.B. Johnson and D.A. Maltz, *Dynamic source routing in ad hoc wireless networks*, Mobile Computing, vol. 353, T. Imielinski and H. Korth, eds., Kluwer Academic, Boston, 1996, Page(s): 153-181.

[43] M.R. Pearlman, and Z.J.Haas, *Determining the optimal configuration for the zone routing protocol*, IEEE Journal on Selected Areas in Communications, Vol.: 17 No. 8 , Aug. 1999 Page(s): 1395-1414.

[44] B. Karp and H. T. Kung, *GPSR: Greedy perimeter stateless routing for wireless networks*, proceedings of the 6th International Conference on Mobile Computing and Networking, ACM MOBICOM 2000, Page(s): 243-254.

[45] C. E. Perkins and P. Bhagwat, *Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers*, proceedings of ACM SIGCOMM 94 Conference on Communications Architectures, Protocols and Applications, Aug. 1994, Page(s): 234-244.

[46] C. E. Perkins and E. M. Royer, *Ad hoc on demand distance vector routing*, proceedings of 2nd IEEE Workshop. Mobile Computing Sys and Apps., Page(s): 90-100, Feb. 1999.

[47] T. Imielinski and J. C. Navas, *GPS-based addressing and routing*, Tech. Rep. LCSR-TR-262, CS Dept., Rutgers University, March (updated August) 1996.

[48] T. C. Hou and T. J. Tsai, *Adaptive clustering in a hierarchical ad hoc network*, proceedings of Int. Computer Symp., Tainan, Taiwan, R.O.C., Dec. 1998, Page(s): 171176.

[49] Y. Wu et.al., *Spreading code assignment in an ad hoc DS-CDMA wireless network*, ICC 2002. Volume: 5, 2002 pp: 3066-3070.

[50] *The ns-2 simulator*, http://www.isi.edu/nsnam/ns (date retrieved: April 7, 2004).

[51] CPLEX optimization Incorporation, *Using the CPLEX Callable Library*, V4.0, Published 1995, http://www.ilog.com/products/cplex/ (date retrieved: April 7, 2004).

[52] M. R. Garey and D. S. Johnson, *Computers and intractability, a guide to the theory of NP-Completeness*, W. H. Freeman and Company, 1979.

[53] D. Goodman, *Wireless personal communications systems*, Reading, MA: Addison-Wesley, 1997.

[54] K. Rosen, *Discrete mathematics and its application*, Chapter 6, Fourth edition, McGraw Hill Publising, 1999.

[55] J. Liu, B. Li. *MobileGrid: Capacity-aware topology control in mobile ad hoc networks*, proceedings of the 11th IEEE ICCCN 2002, Miami, Florida, October 14-16, 2002, Page(s): 570-547.

[56] E. Crawley, R. Nair, B. Rajagopalan, H. Sandrick, *A Framework for QoS based routing in the Internet*, RFC 2386, August 1998.

[57] C. Perkins, E. M. Royer and S. R. Das. *Ad hoc on demand distance vector routing*, Internet-Draft, draft-ietf-manet-aodv-06.txt, July 2000.

[58] D. Johnson and D. Maltz. *Dynamic source routing in ad hoc wireless networks*, In T. Imielinski and H. Korth, editor, Mobile Computing. Kluwer Academic Publ., 1996.

[59] V. Park and M. S. Corson. *A highly adaptive distributed routing algorithm for mobile wireless networks*, In Proc. INFOCOM, Kobe, Japan, April 1997.

[60] H. Xiaoyan, X. Kaixin, M. Gerla, *Scalable routing protocols for mobile ad hoc networks*, IEEE Network, Volume: 16 Issue: 4 , July-Aug. 2002 Page(s): 11-21.

[61] E.M. Royer and C.-K. Toh, *A review of current routing protocols for ad hoc mobile wireless networks*, IEEE Personal Communications (April 1999), Page(s): 4655.

[62] S. Lee and A. T. Campbell. *INSIGNIA: In-band signalling support for QoS in mobile ad hoc networks*, proceedings of the 5th Intl. Workshop on Mobile Multimedia Communication, 1998.

[63] S. Chen and K. Nahrstedt. *Distributed quality of service in ad hoc networks*, IEEE J. Selected Areas in Commun., Vol. 17, No. 8, 1999.

[64] J. N. Al-Karaki, A.E. Kamal, *Quality of service routing in mobile ad hoc networks: Current and future trends*, Mobile Computing Handbook, I. Mahgoub and M. Ilyas (eds.), CRC Publishers, 2004

[65] A. Munaretto, H. Badis, K. Al Agha, G. Pujolle, *A link-state QoS routing protocol for ad hoc networks*, 4th International Workshop on Mobile and Wireless Communications Network, 2002., Sept. 2002, Page(s): 222-226.

[66] Y. Ge, T. Kunz, L. Lamont, *Quality of service routing in ad hoc networks using OLSR*, 36th Annual Hawaii International Conference on System Sciences (HICSS'03), Big Island, Hawaii, 2003.

[67] C. Perkins, E. Royer, SR. Das, *Quality of service for ad hoc on demand distance vector (AODV) routing*, Internet-Draft, July 2000.

[68] R. Leung, J. Liu, E. Poon, A. Chan, B. Li, *MP-DSR: a QoS-aware multi-path dynamic source routing protocol for wireless ad-hoc networks*, proceedings of Local Computer Networks (LCN) 01. Page(s):132-141, 2001.

[69] C. Zhu; M.S. Corson, *QoS routing for mobile ad hoc networks*, procedings of INFOCOM 02. Vol.: 2 , 2002 Page(s): 958-967, 2002.

[70] Y. Chen, Y. Tseng, J. Sheu, and P. Kuo, *On-Demand, Link-State, Multi-Path QoS Routing in a Wireless Mobile Ad-Hoc Network*, Proceedings of Europian Wireless Conference, 2002.

[71] C. R. Lin,*On-Demand QoS Routing in Multihop Mobile Networks*, Proc. IEEE INFOCOM 2001, Volume:3, Page(s): 1735-1744.

[72] S. Shah, K. Nahrstedt, *Predictive location-based QoS routing in mobile ad hoc networks*, ICC 2002, Volume:2, 2002, Page(s):1022-1027.

[73] S.H. Shah, K. Nahrstedt, *Predictive location-based QoS routing in mobile ad hoc networks*, IEEE ICC 2002, Vol.: 2, Page(s): 1022-1027.

[74] J. Chen; J. Wang; S. Deng; Y. Tang; *QoS routing with mobility prediction in MANET*, IEEE Pacific Rim Conference on Communications, Computers and signal Processing (PACRIM.), Vol. 2, 2001. 2001, Page(s): 357-360.

[75] S. Chen, K. Nahrstedt *Distributed quality-of-service routing in ad hoc networks.* IEEE Journal on Selected Areas in Communication Special Issue on Ad hoc Networks, 1999; Vol. 17, No. 8, Page(s): 1488-1505.

[76] G. Raju, G. Hernandez, Q. Zou, *Quality of service routing in ad hoc networks*, IEEE Wireless Communications and Networking Confernce (WCNC) 2000, Vol. 1, Page(s): 263-265, 2000.

[77] M. Hashem, M. Hamdy, S. Ghoniemy, *Modified distributed quality-of-service routing in wireless mobile ad-hoc networks*, MELECON 2002. Page(s): 368-378.

[78] W. H. Liao, Y. C. Tseng, S. L. Wang, and J. P. Sheu, *A multi-path QoS routing protocol in a wireless mobile ad hoc networks*, IEEE international conference on Networking (ICN), 2001.

[79] C. R. Lin and J.-S. Liu, *QoS routing in ad hoc wireless networks*, IEEE Journal on Selected Areas in Communication, vol. 17, No. 8, Aug. 1999, Page(s): 1426-38.

[80] C. Zhu, M.S. Corson,*QoS routing for mobile ad hoc networks*, INFOCOM 2002. Volume: 2 , 2002 Page(s): 958-967.

[81] P. Sinha, R. Sivakumar, V. Bharghavan,*CEDAR: a core-extraction distributed ad hoc routing algorithm*, proceedings of IEEE INFOCOM 99, New York, March 1999.

[82] T. W. Chen, J. T. Tsai, M. Gerla,*QoS routing performance in multihop, multimedia, wireless networks*, IEEE on Universal Personal Communications, Vol.: 2, Page(s): 557-561, 1997.

[83] A. Iwata, C. Chiang, G. Pei, M. gerla, T.W. Chen, *Scalable routing strategies for ad hoc wireless networks*, IEEE Journal Selected Areas in Communications, 17, 8, Aug. 1999, Page(s): 1369-1379.

[84] S. Basagni, I. Chlamtac, V. Syrotiuk, *Dynamic source routing for ad hoc networks using the global positioning system*, IEEE Wireless Communications and Networking Conference, Vol. 1, Page(s): 301-305, 1999.

[85] H. Hartenstein, M. Mauve, A. Widmer, *A survey on position-based routing in mobile ad hoc networks*, IEEE Network, Vol. 15, No. 6 , Nov/Dec 2001, Page(s): 30-39.

[86] S. Capkun, M. Hamdi and J.-P. Hubaux, *GPS-free positioning in mobile ad hoc networks*, Proceedings of Hawaii International Conference on System Sciences, Page(s): 3481-3490, January 2001.

[87] J. Li and P. Mohapatra, *PANDA: A positional attribute-based next-hop determination approach for mobile ad hoc networks*, Technical Report, Department of Computer Science, University of California, Davis, 2002.

[88] M. Maleki, K. Dantu, and M. Pedram, *Power-aware source routing protocol for mobile ad hoc networks*, ACM ISLPED'02, August 12-14, 2002.

[89] S. Singh, M. Woo, C. Raghavendra, *Power aware routing in mobile ad hoc networks*, proceedings of MOBICOM 98 Conference, Dallas, October 1998.

[90] C.K. Toh, *Maximum battery life routing to support ubiquitous mobile computing in wireless ad hoc networks*, IEEE Communication Magazine, June 2001.

[91] K. Chen, S. Samarth and K. Nahrstedt, *Cross layer design for data accessibility in mobile ad hoc networks*, Wireless Personal Communications, Vol. 21, Page(s): 49-76, 2002.

[92] J. Moy. *OSPF Version 2.* INTERNET-RFC 2328, April 1998.

[93] A. Parekh, R. Gallager, *A Generalized processor sharing approach to flow control in integrated services networks: The multiple node case*, IEEE/ACM Transactions on Networking, Vol. 2, No.1, Page(s): 137-150, April 1994.

[94] H. Zhang, *Service disciplines for guaranteed performance service in packet switching networks*, proceedings of the IEEE, Vol. 83, Issue: 10, Page(s): 1374-1396, Oct 1995.

[95] S. Lu, V. Bharghavan, and R. Srikant, *Fair queueing in wireless packet networks*, proceedings of the ACM SIGCOMM Conference (SIGCOMM-97), Page(s): 63-76, 1997.

[96] S. Bucheli, J.R. Moorman, J.W. Lockwood and S. M. Kang, *Compensation modeling for QoS support on a wireless network*, IEEE GLOBECOM 2000, Vol.: 1, Page(s): 198-202.

[97] C. Intanagonwiwat, D. Estrin, R. Govindan, and J Heidemann. *Impact of network density on data aggregation in wireless sensor networks*, proceedings of IEEE ICDCS'02, Vienna, Austria, July, 2002.

[98] B. Krishnamachari, D. Estrin, and S. Wicker, *Impact of data aggregation in wireless sensor networks*, International Workshop on Distributed Event-Based Systems, Vienna, Austria, July 2002.

[99] J. N. Al-Karaki, A.E. Kamal, *A taxonomy of routing techniques in wireless sensor networks*, in Sensor Networks Handbook, M. Ilyas and I. Mahgoub (eds.), CRC Publishers, 2004

[100] K. Sohrabi, J. Gao, V. Ailawadhi, J. Pottie, *Protocols for self-organization of a wireless sensor network*, IEEE Personal Communications, Vol. 7, No. 5, Page(s): 16-27, 2000.

[101] F. Ye, A. Chen, S. Liu, L. Zhang, *A scalable solution to minimum cost forwarding in large sensor networks*, Proceedings of the tenth International Conference on Computer Communications and Networks (ICCCN), Page(s): 304-309, 2001.

[102] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, *Energy efficient communication protocol for wireless microsensor networks,* proceedings of HICSS'00, Jan. 2000.

[103] S. Lindsey, C. Raghavendra, *PEGASIS: power efficient gathering in sensor information systems*, IEEE Aerospace Conference Proceedings, Vol. 3, Page(s): 1125-1130, 2002.

[104] A. Manjeshwar and D. P. Agarwal, *TEEN: a routing protocol for enhanced efficiency in wireless sensor networks*, In 1st International Workshop on Parallel and Distributed Computing- Issues in Wireless Networks and Mobile Computing, April 2001.

[105] A. Manjeshwar and D. P. Agarwal, *APTEEN: A hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks*, proceedings of IEEE IPDPS 2002, Page(s): 195-202.

[106] V. Rodoplu and T. H. Meng, *Minimum energy mobile wireless networks*, IEEE Journal Selected Areas in Communications, vol. 17, No. 8, Aug. 1999, Page(s): 1333-44.

[107] S. Bandyopadhyay, E. Coyle, *An energy efficient hierarchical clustering algorithm for wireless sensor networks*, proceedings of INFOCOM 2003, Vol. 3, Page(s): 1713-1723.

[108] L. Li, and J. Y. Halpern, *Minimum energy mobile wireless networks revisited,* IEEE International Conference on Communications (ICC) 2001. Vol. 1, Page(s): 278-283.

[109] Y. Xu, J. Heidemann, D. Estrin, *Geography-informed energy conservation for ad hoc routing*, proceedings of the Seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking 2001, Page(s): 70-84.

[110] Q. Li and J. Aslam and D. Rus, *Hierarchical power aware routing in sensor networks*, proceedings of the DIMACS Workshop on Pervasive Networking, May, 2001.

[111] F. Ye, H. Luo, J. Cheng, S. Lu, L. Zhang, *A Two-tier data dissemination model for large-scale wireless sensor networks*, proceedings of ACM/IEEE MOBICOM, 2002.

[112] W. Heinzelman, J. Kulik, and H. Balakrishnan, *Adaptive protocols for information dissemination in wireless sensor networks*, proceedings of 5th ACM/IEEE MOBICOM 99, Seattle, WA, August, Page(s): 174-185, 1999.

[113] J. Kulik, W. R. Heinzelman, and H. Balakrishnan, *Negotiation-based protocols for disseminating information in wireless sensor networks*, Wireless Networks, Vol.: 8, Page(s): 169-185, 2002.

[114] J.-H. Chang and L. Tassiulas, *Maximum lifetime routing in wireless sensor networks*, proceedings of Advanced Telecommunications and Information Distribution Research Program (ATIRP2000), College Park, MD, Mar. 2000.

[115] C. Rahul, J. Rabaey, *Energy aware routing for low energy ad hoc sensor networks*, IEEE Wireless Communications and Networking Conference (WCNC), Vol.1, Page(s): 350-355, March 17-21, 2002.

[116] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, *Highly-resilient and energy-efficient multipath routing in wireless sensor networks*, ACM SIGMOBILE Mobile Computing and Communications Review, Vol. 5, No. 4, Page(s): 1125, 2001.

[117] D. Braginsky and D. Estrin, *Rumor routing algorithm for sensor networks*, proceedings of International Conference on Distributed Computing Systems (ICDCS'01), November 2001.

[118] B. Chen, K. Jamieson, H. Balakrishnan, R. Morris, *SPAN: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks*, Wireless Networks, Vol. 8, No. 5, Page(s): 481-494, September 2002.

[119] J. N. Al-Karaki, R. Ul-Mustafa, A. E. Kamal, *Data aggregation in wireless sensor networks: Exact and approximate algorithms*, proceedings of IEEE Workshop on High Performance Switching and Routing (HPSR), April 18-21, 2004, Phoenix, Arizona, USA.

[120] Raza Ul-Mustafa, *Design and provisioing of WDM networks for traffic grooming*, ECpE department, ISU, Ph.D. Thesis.

[121] A. Savvides, C-C Han, aind M. Srivastava, *Dynamic fine-grained localization in ad hoc networks of sensors*, MOBICOM 01), Page(s): 166-179, July 2001.

[122] S. Madden, M. Franklin, J. Hellerstein, and W. Hong, *TAG: a Tiny aggregation service for ad hoc sensor networks*, OSDI 2002, Boston MA, December 2002.

[123] D. E. Goldberg, *Genetic algorithms in search, optimization, and machine learning*, Edison Wesley, 1989.

[124] S. Z. Selim and M. A. Ismail, *k-Means type algorithms: A generalized convergence theorem and characterization of the local optimality*, IEEE Transactions on Pattern Analysis, Vol.6, No.1, Page(s): 81-86, 1984.

[125] S. Singh and C.S. Raghavendra. *PAMAS: Power aware multi-access protocol with signalling for ad hoc networks*, ACM Computer Communication Review, Vol. 28, No. 3, Page(s): 5-26, July 1998.

[126] W. Ye, J. Heidemann, and D. Estrin, *An energy-efficient MAC protocol for wireless sensor networks*, proceedings of INFOCOM 02, Vol.: 3, Pages: 1567-1576, 2002.

[127] G.J. Pottie and W. J. Kaiser. *Wireless integrated network sensors*, Communications of the ACM, Special Issue on Embedding the Internet, Vol. 43, No. 5, Page(s): 5158, May 2000.

[128] S. Shakkottai, R. Srikant and N. Shroff, *Unreliable sensor grids: Coverage, connectivity and diameter*, proceedings of IEEE INOCOM 03, San Francisco, CA, April, 2003.

[129] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. Srivastava, *Coverage problems in wireless ad hoc sensor networks*, proceedings of IEEE INFOCOM'01, Anchorage, AK, 2001.

[130] L. Blazevic, L. Buttyan, S. Capkun, S. Giordano, J. P. Hubaux, J. Y. Le Boudec, *Self-Organization in Mobile Ad-Hoc Networks: the Approach of Terminodes*, IEEE Communications Magazine, Vol. 39, No. 6, June 2001.

[131] S. Marti, T. J. Giuli, K. Lai, M. Baker, *Mitigating routing misbehavior in mobile ad hoc networks*, proceedings of MOBICOM 00 Boston-MA, August 2000.

[132] N. Ben Salem, L. Buttyan, J. P. Hubaux, and M. Jakobsson, *A charging and rewarding scheme for packet forwarding in multihop cellular networks*, proceedings of MobiHoc 2003.

[133] L. Buttyan, J. P. Hubaux, *Stimulating cooperation in self organizing mobile ad hoc networks*, in ACM/Kluwer Mobile Networks and Applications (MONET), Vol. 8, No. 5, Oct. 2003.

[134] L. Anderegg, S. Eidenbenz, *Ad hoc-VCG: A truthful and cost-efficient routing protocol for mobile ad hoc networks with selfish agents*, International Conference on Mobile Computing and Networking, Pages: 245-259, 2003.

[135] Stanford University Mobile Activity TRAces, *http://www-db.stanford.edu/sumatra/*, SUMATRA trace generator, (date retrieved: January 2, 2004).