2005

Performance evaluation of hierarchical ad hoc networks.

Tai Yu
University of Windsor

Performance Evaluation of Hierarchical Ad Hoc Networks

by

Tai Yu

A Thesis
Submitted to the Faculty of Graduate Studies and Research through the Department of Electrical and Computer Engineering in Partial Fulfillment of the Requirements for the Degree of Master of Applied Science at The University of Windsor

Windsor, Ontario, Canada

Your file Votre référence
ISBN: 0-494-04974-X
Our file Notre référence
ISBN: 0-494-04974-X

Abstract

Ad hoc networking is one of the most challenging areas of wireless communication. Theoretical analysis and experimental results show that QoS (Quality of Service) for each node degrades rapidly while the number of nodes increases in the network. One way to solve performance degradation is to use hierarchical network architectures.

In this paper, we investigate performance improvements offered by hierarchical ad hoc networks over flat (non-hierarchical or conventional) ad hoc networks for QoS parameters, namely throughput capacity, delay and power efficiency. We investigated and identified trade-offs among those QoS parameters via computer simulations carried by Network Simulator 2 of University of California (NS-2). In those simulations, we created hierarchical ad hoc networks by clustering the networks using cluster head nodes. Initially network is static (no mobility). Results of static network simulations act as benchmark for the performance parameters. Later mobility scenarios are added into the network to observe how mobility affects the performance. In order to compare two architectures, hierarchical and flat, we systematically changed number of nodes, data packet generation rates, number of clusters, node densities and transmission ranges for the nodes. At the same time, we compared hierarchical ad hoc network architecture with WLAN architecture, which has full infrastructure.

Simulation results state that throughput performance is linear with numbers of clusters; and in hierarchical architecture, power efficiency is doubled and delay is significantly lower than flat architecture. Our simulation results conclude that clustering schemes in wireless ad hoc networks can solve the scalability problem that exists in flat architectures.

. . .

路漫漫其修远兮；

吾将上下而求索！

. . .

《离骚》，屈原， 公元前 343-289

..

Long long is the way, but nothing'll my effort arrest;

Up hill and down dale for the beauty I will quest!

...

Qu Yuan (343 BC-289 BC), "Tales of Woe"
Greatest poet of ancient China
Remembered every year by holding dragon boat races

Acknowledgments

I wish to express sincere appreciation to my supervisor Dr. K. Tepe, very few professors will take such care in students' research, and to Dr. Khalid and Dr. Jaekel for their advices in my research process and assistance in the preparation of this manuscript. In addition, I would like to thank Tarique and Naserian, who are my buddies in the wireless ad hoc research group, for their suggestions and pep talk.

I would like to sincerely thank my wife Fei, who takes care of my family and her cookery makes our life colorful, although she is busy with her work as well. And, I am eternally thankful for the love of my mother and father. They always find a way to make me proud of them and encourage me in the research for science.

At the end, I want to thank the actual reader, because, reading this thesis gives a sense to my work.

Table of Contents

List of Figures

Number *Page*

List of Tables

Number *Page*

Abbreviations

ACK: Acknowledgment

AODV: Ad hoc On-Demand Distance Vector

AP: Access Point

BS: Base Station

CBR: Constant Bit Rate

CH: Cluster Head

CTS: Clear to Send

CBR: Constant Bit Rate

DSDV: Destination Sequenced Distance Vector

DSR: Dynamic Source Routing

DSSS: Direct Sequence Spread Spectrum

F-MANET: Flat Mobile Ad-hoc Networks

H-MANET: Hierarchical Mobile Ad-hoc Networks

MANET: Mobile Ad-hoc Networks

MN: Mobile Node

NOAH: NO Ad-Hoc Routing Agent

NS-2: Network Simulator Version 2

OTcl: Object-oriented Tool Command Language

QoS: Quality of Service

RTP: Random Traffic Pattern

RTS: Request to send

RW: Random Waypoint

TCL: Tool Command Language

WLAN: Wireless Local Area Network

Chapter 1

Introduction

The history of wireless networks started in the 1970s and the interest has been growing ever since. Wireless network has been becoming more and more important in our daily life, from WLAN to cellular phone systems. To evaluate a wireless system's performance, we usually look into QoS, whose key parameters include throughput capacity of the each Mobile Node (MN) and delay of the each communication link. Because wireless device is powered by battery, power efficiency is also a very important parameter, and total power consumption determines the survival time of the networks.

1.1 Wireless Network Architectures

Today we see many kinds of wireless networks but the difference between them is not as obvious as it may seem. Wireless networks are classified into two categories by architectures: infrastructured and infrastructureless.

For infrastructured architecture, we can at least split the network into two tiers, the nodes in different tier have different functionalities. Lower tier nodes are fully controlled by upper tier nodes such as WLAN. For infrastructureless architecture, connections between two nodes are peer to peer, where there are no controlling nodes.

1.2 MANETs

The typical example of peer to peer wireless network is mobile ad-hoc networks [MANETs]. They can be set up anywhere and anytime because they eliminate the complexities of infrastructure setup, central administration.

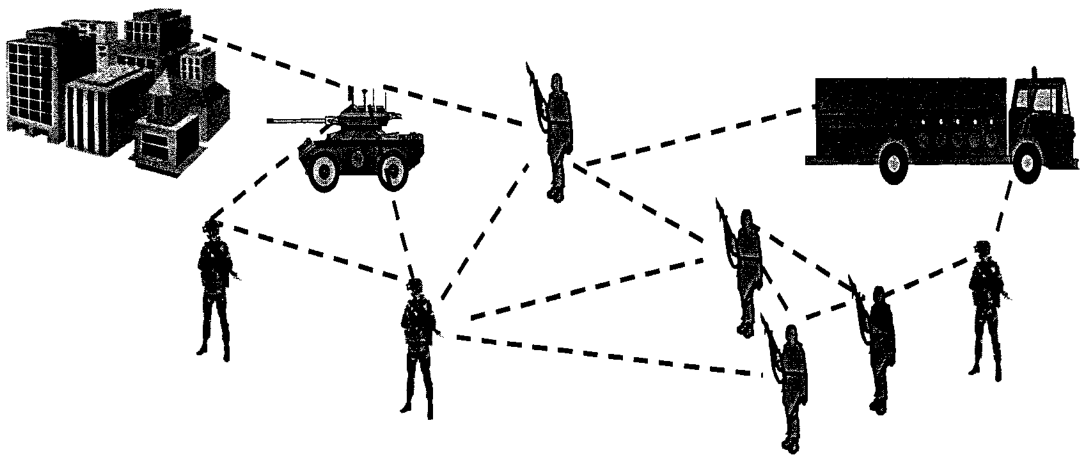

Figure 1.1 MANET

The MANETs came from US army in some applications, for example, soldiers in a battlefield need to communicate among themselves, as shown in Figure 1.1. The word ``ad hoc" can be translated as ``not organized" which often has a negative meaning, but in our context it is not negative describing the network situation (i.e., dynamic). All or some nodes in an ad hoc network are expected to be able to route data-packets for other nodes in the network that want to reach other nodes beyond their own transmission range, which is called peer-level multi-hopping. MANETs can also be applied to other applications; for example, MANETs can be used in data acquisition of sensor networks, or in the aftermath of a natural disaster such as earthquake where cellular service may not be available.

In a cell-based wireless network such as WLAN, nodes communicate with each other through AP (Access Point), each AP controls a node inside its coverage area, named cell. A node first connects to the nearest AP to communicate with other nodes. Because an AP serves as a communication gateway and control center for all the nodes in its cell, infrastructure-based architecture is more reliable and has better QoS performances. However, MANET is more desirable because of its low cost, plug-and-play convenience, and flexibility. MANET's usage of bandwidth and battery power is more efficient if we keep the number of nodes in a network small.

1.3 Problem Statement

Throughput capacity is a key characteristic of wireless networks. It represents the long-term achievable data transmission rate that a network can support. Gupta and Kumar in [1] showed that the average available throughput per node decreases with the square root of the number of nodes n in the network. In a static MANET, the throughput capacity is at most in the order of $\theta\left(\frac{W}{\sqrt{(n \log n)}}\right)$, where W is the channel bit rate. Equivalently, the total network capacity increases as at most $\sqrt{n}$. In particular, that holds irrespective of he network topology, power control policy or any transmission scheduling strategy. Figure 1.2 shows that throughput obtainable by each node decreases quickly if we keep increasing number of nodes in the system.

Gupta and Kumar [1] did not consider the delay constraints and offer solutions in the network. But they pointed out that enhancing the network throughput capacity would be at the expense of increased transmission delay. Some methods are taken to enhance the capacity performance of MANETs. For example, Comaniciu and Poor [5] investigated the asymptotic capacity for delay sensitive traffic in ad hoc networks with signal processing techniques such as multiuser detection for capacity performance. Bansal and Liu [2] proposed a routing algorithm for MANETs with the goal of achieving optimal capacity while keeping the delay small. The algorithm exploits the patterns in the mobility of nodes to provide guarantees on the delay. However, as noted in [2] their routing-specific scheme does not provide a general solution to guarantee the MANETs' QoS with a wide range of mobility patterns of users. Neely and Modiano[4] considered the throughput and delay tradeoffs for scheduling data transmission in a MANET. Delay analysis for the MANET was facilitated using a simple mobility model. Based on the above analysis, if the network size gets large, the delay increases at least in the order of $O(\log(n))$, where n is the number of nodes in the system. In order to reduce the high energy consumption in MANETs, the authors in [6] and [7] investigated the power consumptions in MANETs and proposed some graph-theoretic based algorithms in routing to control topology using transmit power adjustment.

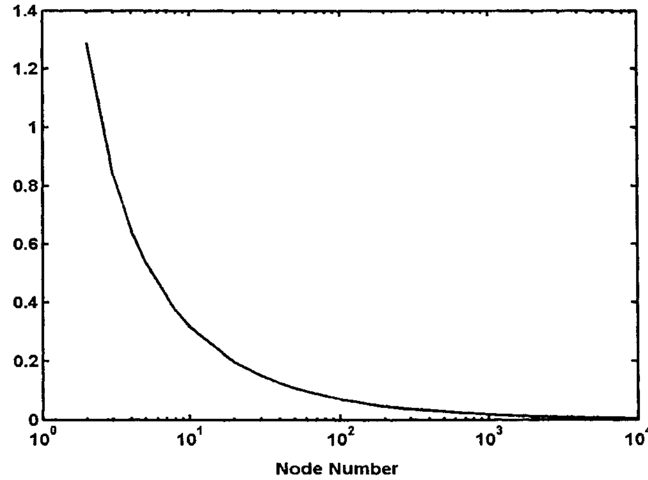

Figure 1.2 Obtainable Throughput by Each Node in MANET

Following properties can be concluded for homogeneous MANETs, or as we call it Flat MANET (F-MANET). When a F-MANET grows larger: 1) Throughput degrades 2) end-to-end delay increases 3) power efficiency drops. Based on the above analysis, we conclude that MANETs do not scale! However, all the above research just stated the problem in one aspect existing in F-MANET architecture, but none of the above research presents good solutions to solve the problem of scalability.

1.4 Methods

Due to the weak scaling capability, we are considering introducing infrastructure support to F-MANETs. While the F-MANET provides better coverage and flexibility, the infrastructure-based architecture provides better scalability in terms of throughput and delay performance. However, neither of those has simultaneous flexibility and scalability.

Combining those two architectures, results in a hybrid architecture called Hierarchical ad hoc networks (H-MANET). H-MANET presents a tradeoff between WLAN and F-MANET so that data may be forwarded in a multi-hop fashion or through the

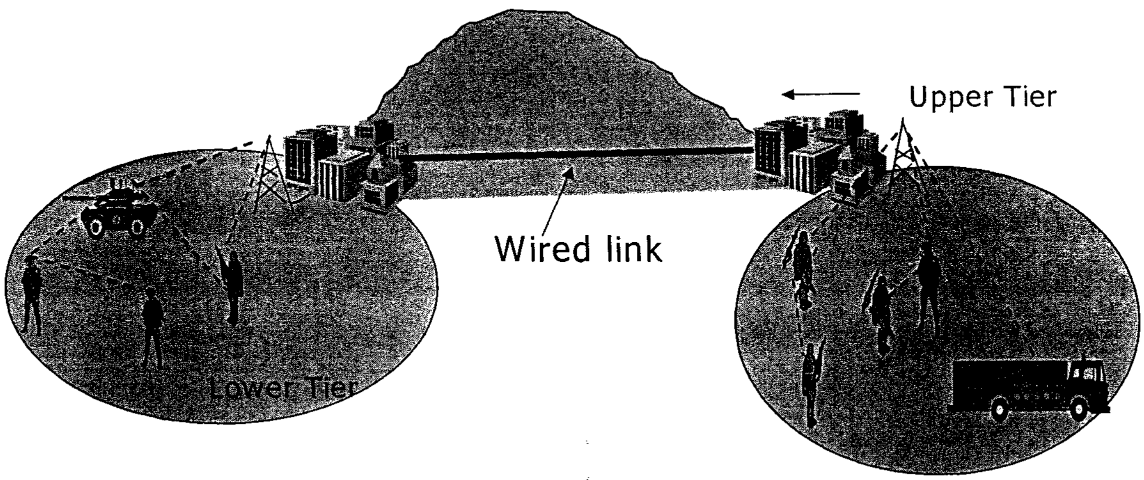

Fig1.3 H-MANET

infrastructure.

In order to change the architecture of F-MANET in Figure 1.1 to the Figure 1.3, we cluster the F-MANET into small sections, where each section is called a cluster. The behavior of a communication inside a cluster will be as same as F-MANET, and we assign an AP to each cluster. In order to distinguish the AP from WLAN's AP, we call the H-MANET's AP as "Cluster Head (CH)". Here, we split the network into 2 tiers, where lower tier nodes are mobile nodes (MNs) and upper tier nodes are CHs connected by wired link. If the source and destination are not located in a same cluster, it will be routed through upper layer. The behavior in this case is somewhat like WLAN; however, we can see at least two differences between WLAN and H-MANET. First, for intra-cluster traffic in H-MANET, no CH will be involved in the communication. Second, for inter-cluster traffic in H-MANET, MNs can reach their cluster head in multi-hop. Based on these two characteristics of the H-MANET architecture, we try to achieve a network combining F-MANET's flexibility and WLAN's scalability.

1.5 Research Objective

In the literature, authors did not investigate the interaction of different network layers on the network performance. It is hard to analyze and obtain a closed formula that captures effect of all the layers on the performance figures. That is why in this thesis, we will use a simulation approach to investigate QoS performance of hierarchical and flat ad hoc networks, and compare those to the currently available networks such as WLAN. [8] and [9] analyzed the throughput performance of a very simple hybrid architecture; for example, they assumed stationary network and no power and delay constraints. Based on our knowledge, no delay performance analysis for H-MANET architecture can be found.

The goal of our research is to systemically investigate the performance of H-MANET architecture, from which we are able to analyze the benefits of the infrastructure to QoS, and to find out the inter-changing of key parameters simultaneously. Results of this research will enable us to broaden our knowledge about H-MANET architecture and give us information if it really provides QoS improvement over F-MANET. At the same time, we will compare this new architecture with WLAN and find if it offers any advantages over WLAN.

Our simulations explored the flat and hierarchical wireless ad hoc network architectures via computer simulations. We investigated QoS parameters, namely throughput capacity, average packet delay and power efficiency. For throughput performance, we tested the maximum throughput a system can reach and what percent of throughput can be successfully delivered to their destinations (i.e., delivery ratio). For delay performance, we looked into each communication link and took an average. For power analysis, we calculated the total energy consumption and then calculate the power efficiency based on the throughput performance.

Chapter Summary:

Clustering schemes in wireless ad hoc networks have been investigated in our study to enhance network manageability and channel efficiency, and to solve the scalability problem in flat architecture. In this thesis, we focus on throughput, delay and power efficiency improvements made possible by clustering. To achieve our goal, we systematically investigate the changing of these parameters in H-MANET architectures and find the relationships among these parameters in the following chapters, from which, we know the benefits offered by clustering schemes, which gives us some ideas how to properly design an ad hoc network with infrastructure support.

Chapter 2

System Models

In this chapter, first, we will describe the protocol stack used in our simulations. Second, we will build our system models with F-MANET and H-MANET architectures. For each architecture, hop number will be used in simulation model to control topology. After that, a simple mathematical model is presented to help us analyze scalability problems in F-MANET.

2.1 Protocol Stack

The protocol stack that we use in our study model is an IEEE 802.11 based network. For the analysis of network performance, we look into bottom two layers with CBR application on the top. We have two node types in our system, MN and CH. The protocol stack for two node types are shown in Figure 2.1

2.1.1 Routing Protocols

Routing protocol provides forwarding packets from source to destination along the most optimum route. Usually, routing protocols are adaptable and offer multi-hop paths across a network. Traditional routing protocols were designed for infrastructure-based networks, which usually are one hop allowed in wireless medium. While ad hoc network routing protocols comprise a complex and active field of research, and can be broadly classified into two categories.

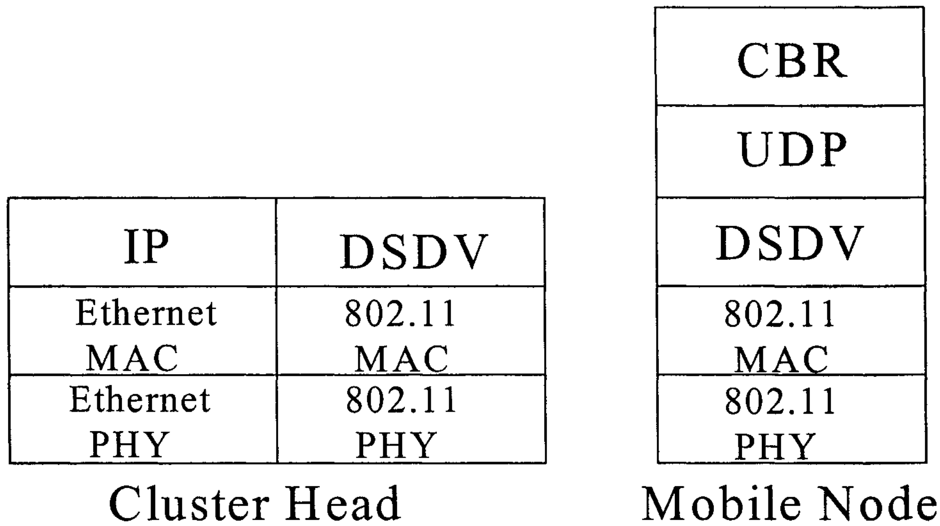

Figure 2.1 Protocol Stack in the System Models

1) Proactive or Table-Driven Routing

This kind of routing protocols record routes for all destinations in the network, which is based on traditional wired routing protocols. The routing information is disseminated among all nodes in the network throughout the operating time irrespective of the need for such a route.

Advantage: Communications with arbitrary destinations experience minimum initial delay since routes can be immediately selected from routing table.

Disadvantages: Additional control traffic is needed to continuously update stale route entries. Unlike the wired network, an ad hoc network contains mobile nodes and therefore links are continuously broken and reestablished.

Proactive or table-drive routing protocols can be subdivided depending on how the routing tables are constructed, maintained and updated [10]. Proactive routing is "link state" routing protocol. In this scheme, each node maintains a view of the entire network topology with a cost for each link. To keep these views consistent, each node periodically broadcasts the link costs of its outgoing links to all other nodes using a protocol such as flooding. As a node receives the information, it updates its view of the network topology and applies a minimum-cost algorithm to choose its next hop for each destination. The second one is “distance vector”. In distance vector routing, each node maintains a routing table consisting of a destination address, distance to the destination and the next node in the path. Each router periodically broadcasts this table information to each of its neighboring routers, and uses similar routing updates received from its neighbors to update its own table. DSDV is the most popular proactive routing protocol used in MANETs. DSDV is based on idea of classical “distance vector” routing algorithm. The attributes of each destination are the next hop, the number of hops to reach to the destination, and a sequence number, which is originated by the destination node. Table 2.1 addresses the means taken by DSDV to solve traditional problems in “distance vector”

Table 2.1 Comparison of DSDV and DV

Problems of "Distance Vector"	Solutions in "DSDV"
Topology changes are slowly propagated, or Count-to-infinity problem.	Tagging of distance information: •The destination issues increasing sequence number •Other nodes can discard old/duplicate updates
Moving nodes create confusion, They carry connectivity data which are wrong at new place.	Changes are not immediately propagated Wait some setting time in DSDV
Table exchange eats bandwidth	Incremental updates instead of full table exchange

In DSDV routing protocol, both periodic and triggered routing updates are used to maintain table. When a node receives an infinity metric with a later sequence number, it will trigger a route update broadcast to the disseminate the news.

2) Reactive or On-Demand Routing

In On-Demand routing scheme, the nodes do not need to update their routing tables periodically. Since a node in an ad hoc network does not need a route to a destination until that destination is to be a recipient of packets sent by the node.

Advantage: Uses far less bandwidth to maintain route table at each node at slow networks.

Disadvantage: Since the route to a destination will have to be acquired before communications can begin (route discovery), the latency period for most applications is likely to increase drastically.

The mostly used On-Demand Routing protocols in MANETS are DSR and AODV. However, DSR and AODV do not support clustering scheme in our simulation environment. That is why we use DSDV in our flat and hierarchical architecture to evaluate the performance of F-MANETs and H-MANETs.

2.1.2 MAC Protocols

No particular MAC layer protocols were designed for MANETs so far. In our study, we use 802.11[11] to MANETs in our simulations. In 802.11, in order to reduce collisions caused by hidden terminals in wireless networks, a four-way exchange scheme RTS/CTS/DATA/ACK is applied to MANETs. In brief, a node that needs to send a data packet first sends an RTS to the destination. If the destination believes that network is idle, it responds with a CTS packet. The sender then transmits the data packet, and waits for an ACK from the receiver. If a node overhears an RTS or CTS, it knows the medium will be busy for some time, and avoids initiating new transmissions or sending any CTS packets.

802.11 RTS and CTS packets include the amount of time the medium will be busy for the reminder of the exchange. Each node uses these times to update its network allocation vector (NAV). The NAV value indicates the amount of time remaining before the network will become available. Upon successful receipt of an RTS frame not addressed to itself, a node updates its NAV to the maximum of the time carried in the RTS frame and its current NAV value. Upon receiving an RTS addressed to itself, a node returns a CTS frame only if its NAV value is zero, otherwise no CTS is sent. Hense, a sender will see no CTS if its RTS packet has collided with another transmission at the receiver, or if the receiver's NAV indicates that the network is not available. A node times out and re-sends the RTS if it receives no CTS.

802.11 doubles the backoff window each time a timeout occurs; it resets the backoff to a minimum value after successful transmission or a transmission is dropped after reaching maximum retry limit.

2.1.3 PHY Layer

The IEEE comprises several alternative physical layers that specify the transmission and reception of 802.11 frames. In our system models, 802.11b Physical Layer was employed, which uses direct sequence spread spectrum (DSSS) technology to support operation of up to 11Mbps data rates in the 2.4GHz band. In our simulations, data rate 2Mbps is used.

Physical layer is network interfaces in the system model. The Network interface layer serves as a hardware interface, which is used by MN to access the channel. This interface subject to collisions and the radio propagation model receives packets transmitted by other node interfaces to the channel. The interface stamps each transmitted packet with the meta-data related to the transmitting interface like the transmission power, wavelength etc. The model approximates the DSSS radio interface of LucentWaveLan Card in our simulation model. An omni-directional antenna having unity gain is used by MNs.

The radio propagation model uses Friss-space attenuation ($1/r^2$) at near distances and an approximation to Two-ray-Ground ($1/r^4$) at far distances, which assumes specular reflection off a flat ground plane.

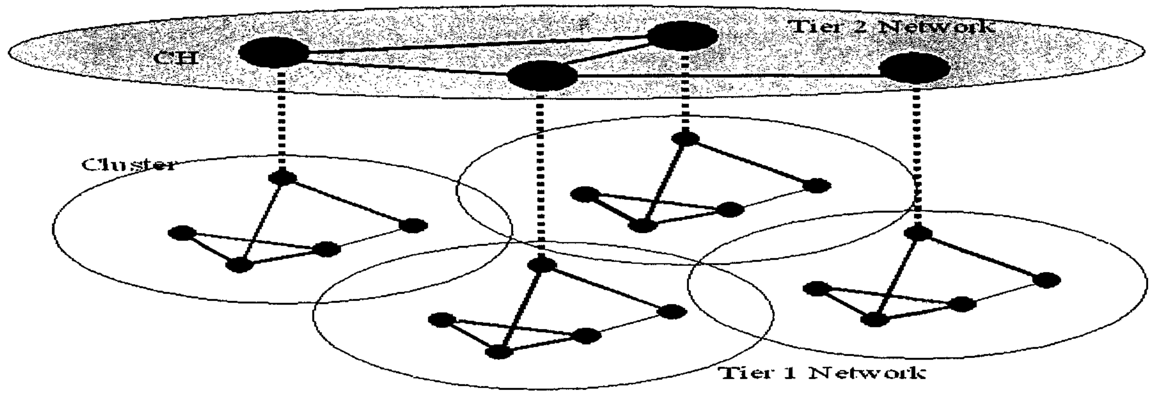

Figure 2.2 Cluster Scheme in F-MANET

2.2 Architectures and Hop Control

As we discussed in the first chapter, a scalable architecture for wireless communications via ad hoc networks has been proposed as H-MANET. Although its implementation in hardware and software has yet to be developed, the performance can be analyzed by simulation on system models. We can use clustering techniques to construct the new architecture. As shown in Figure 2.2, one of the basic principles of H-MANET architecture is that of a hierarchy of networks built on local area networking via F-MANET, with enough flexibility to provide service on demand and to enable growth as the need arises [12]. In the proposed architecture, CHs would serve as relay stations for other MNs, enabling any MN within the range of another, to gain access to the local- and wider-area networks in a multi-hop manner. For the better understanding of the new architecture, we construct the system models based on complexities, which are F-MANET system model, static one-hop H-MANET, static multi-hop H-MANET and mobile multi-hop H-MANET. We will also construct a WLAN system model for the comparison with above systems. Number of hops is controlled by transmission range of the MNs.

2.2.1 F-MANET System Model

In order to investigate how much improvement can be obtained on the QoS of H-MANET in terms of throughput, delay and power efficiency, we simulated F-MANET architecture for comparison.

The following characteristics are identified of F-MANETs. Firstly, new members can join and leave the network any time. Secondly, no CH to provide connectivity to backbone hosts or to other MNs. Finally, no need for handover and location management, as each MN acts as a router, forwarding packets from one MN to another. If we remove CHs form Figure 2.2, the network becomes a F-MANET. The top view of the system is shown in Figure 2.3.

The F-MANET networks with same configurations as corresponding H-MANET will be constructed in our simulation, which means F-MANET networks have the same coverage areas, node number, transmission range or mobile scenarios. Based on these assumptions, it is fairly clear to see the architecture's impact on QoS.

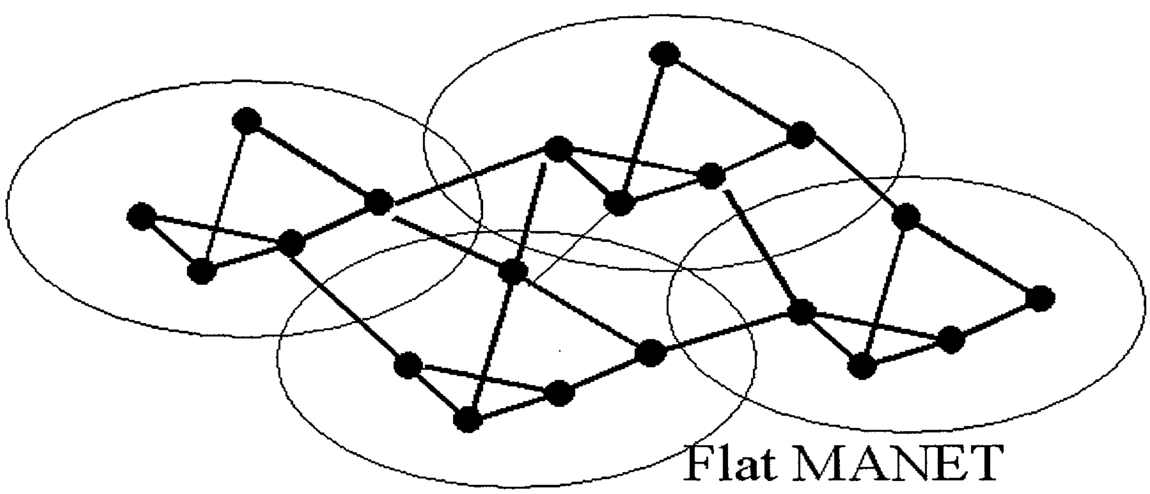

Figure 2.3 F-MANET System Model

2.2.2 Static One-hop H-MANET

Static one-hop H-MANET is the simplest network in H-MANET. Note that in most mobility scenarios, MNs do not move significant distances during packet transit times. Thus, for capacity analysis, we can view mobile networks as effectively static. In our system model, network is clustered with small square clusters or disk clusters with radius

R. A typical system model with 4 clusters is shown in Figure 2.4, where MNs are randomly placed. Node number, cluster numbers are varied for different simulation purposes.

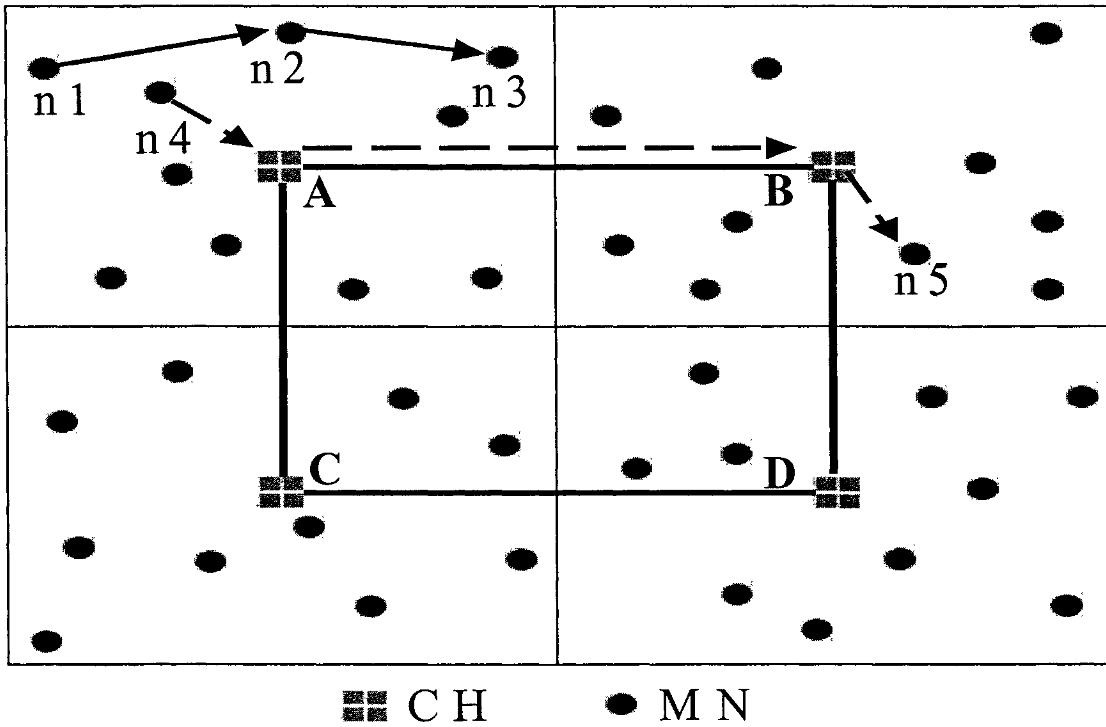

Figure 2.4 Static One-Hop H-MANET

In the system model, network is split into two tiers. Upper tier consists of CHs linked with wired LAN, and each CH is centered in the cluster, in that MNs can reach its CH in one hop. Connection patterns are randomly chosen; therefore, there is 75% inter-cluster traffic and 25% intra-cluster traffic is employed in a four-cluster scheme. In Figure 2.4, packets between *n4* and *n5* (*n4* and *n5* are MNs and not located in same cluster) are routed through upper tier network. Packets between *n1* and *n3* are routed through *n2* as they should be in F-MANET, without any CH being involved.

2.2.3 Static Multi-hop H-MANET

In static multi-hop H-MANET system model, a MN can reach its CH in many hops. An example system model is shown in the Figure 2.5:

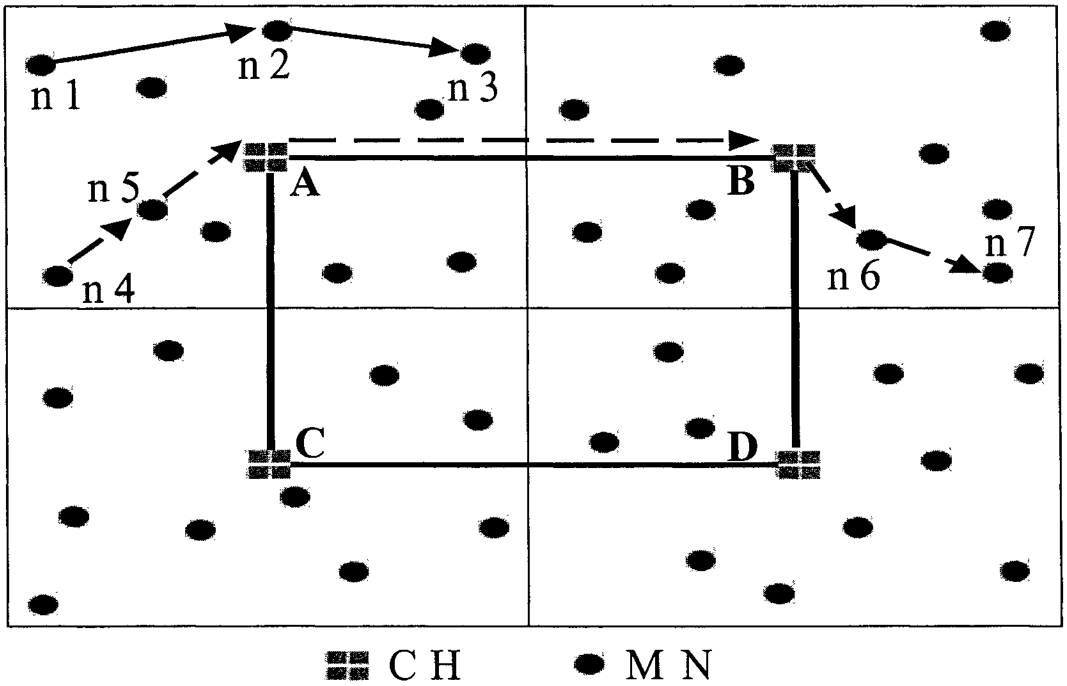

Figure 2.5 Static Multi-Hop H-MANET

For example, link between *n4* and *n7* can be established in this system model, which fails in one-hop system. Intra-cluster traffic still has the same behaviors of F-MANET. Basically, this system extends the geographical coverage area of H-MANETs by allowing multi-hop connections.

2.2.4 Mobile Multi-hop H-MANET

In our system model, we assume there are no handoffs or roaming taking place when a MN is moving. Our mobility scenarios are applied to the multi-hop system and evaluate the mobility's impact on this system model.

The Random Waypoint model [15][16] is the most commonly used mobility model in the research community. The mobility model is as follows: at every instance, a node randomly chooses a destination and moves towards it with a velocity chosen uniformly randomly from [0, V], where V is the maximum allowable velocity for every mobile node.

After reaching the destination, the node stops for a duration defined by the 'pause time' parameter. After this duration, the node again chooses a random destination and repeats the whole process again until the simulation ends.

We fixed maximum velocity to a reasonable value, such as 60km/hr. The system becomes less and less dynamic with increasing "pause time".

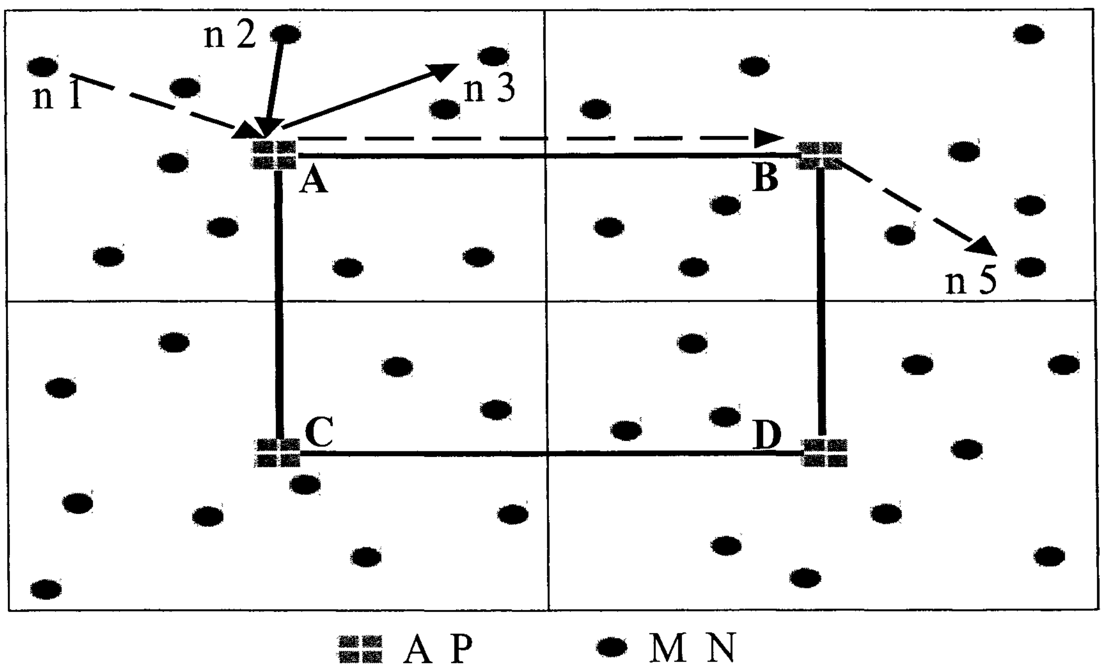

Figure 2.6 WLAN System Model

2.2.5 WLAN System Model

H-MANET is an architecture presenting a trade off between F-MANET and WLAN. We investigate the WLAN network with same configurations as H-MANET in order to fairly compare H-MANET with the currently available wireless networks.

The WLAN has the architecture as shown in Figure 2.6. For any connections, data packets have to go through AP, even if the source and destination are located in one cluster. Transmission ranges of APs need to be increased to reach all the nodes in the cluster to avoid any isolated MNs.

2.3 Mathematical Analysis of Capacity

In this section, we analyze the throughput performance with random traffic pattern (RTP). RTP is the most common traffic pattern used in simulations of ad hoc networks. In RTP, each source node initiates packets to randomly chosen destinations in the network.

First, let us derive the expected path length $\overline{L}$ for such traffic. Since a node chooses every node as its destination with equal probability, the probability that a node *Y* chooses a destination within *R* distance away is proportional to the number of nodes in the disc centered with *Y* and radius *R*. We assume that there is no boundary effect involved in our analysis. When the node density is constant, the number of nodes is proportional to the area of the disc, that is

$$n \propto A \propto R^2 \qquad (1)$$

where *n* is the node number in the network and *A* is disk area. This is an un-normalized cumulative distribution function of the probability of a node communicating with a node at most certain distance units away. We know the maximum distance is 2*R* for a disk network with radius *R*. Taking the derivative of the cumulative distribution function and normalizing it, we get the probability density function giving the probability of a node communicating with another node at distance x as:

$$p(x) = \frac{x}{\int_0^{2R} t\,dt} \qquad (2)$$

where A is the area of the disk. Therefore, from Equation (2), the expected path length for a random traffic pattern is

$$\overline{L} = \int_{0}^{2R} x p(x) dx = \frac{4R}{3} \qquad (3)$$

Substitute *R* with n form Equation (1), we have

$$\overline{L} \propto \sqrt{n} \qquad (4)$$

Assuming that the node density γ is uniform, we have $\gamma = n/A$. Therefore, the total one-hop capacity of the network *C*, should be proportional to the area, or $C = kA = k^* n/\gamma$ for some constant *k*. If we assume the transmission range of a node is *r*, we have capacity obtainable by each node *Cn*, is bounded by

$$C_n < \frac{C/n}{\overline{L}/r} = \frac{kr}{\gamma} \times \frac{1}{\overline{L}} \qquad (5)$$

Because *k*, *r* and γ are constant values in this case, we can bound each node's throughput $C_n \propto \frac{1}{\overline{L}}$, from Equation 4, we can bound it as

$$C_n \propto \frac{1}{\sqrt{n}} \qquad (6)$$

Equation 6 demonstrates an upper bound of obtainable capacity by each node in traditional ad hoc networks. In the real simulation, the result will be lower than this value, because each node's radio transmission range needs to increase in order for an ad hoc network to stay connected with high probability as the number of nodes increases. Once again, Equation 6 clearly shows the scalability problem of F-MANET networks.

Chapter Summary:

In this chapter, we introduced the protocol stack and different system models, which will be followed in our simulations. From the mathematical analysis, we know that F-MANET network does not scale and we designed four simulation models to evaluate the performance of H-MANET architecture.

Chapter 3

Simulation Tool and Modifications

Network Simulator (NS) [17] is a very popular simulation tool in academic research field due to its open source. NS is an event driven network simulator developed at UC Berkeley that simulates variety of networks. There are over 10 thousand users around the world in over one thousand institutions. In [18], NS-2 and most popular commercial simulation tool OPNET were compared with a network testbed. The accuracy of NS-2 and Modeler from OPNET was compared using CBR data traffic and an FTP session. Several scenarios were evaluated and regenerated in the simulation tools and the network testbed. From the researchers point of view, NS-2 provides very similar results compared to OPNET Modeler, but the "freeware" of NS-2 makes it more attractive to researchers.

We use NS-2 to simulate the ad hoc network architectures. However, some desirable functionalities are not provided in NS-2 for our evaluation; for example, NS-2 does not provide simulation environment for H-MANET and WLAN architectures, which are desirable in our comparison. Moreover, some parameters such as delay and power consumption are not provided by NS-2. We modified NS-2 simulation environment in order to implement H-MANET architecture and evaluate its performances.

3.1 Network Simulator 2

In our simulation models, NS-2 implements network protocols such as UPD, traffic source behavior such as CBR, router queue management mechanism such as drop tail, routing algorithms such as DSDV. Basically, NS-2 runs on all Unix platforms and some versions run on MS Windows. All the simulations in our report were run on Linux, an open source OS. The NS-2 introduction (Section 3.1) is reproduced from *NS by Example* [19]. Detailed information of NS-2 can also be found in [17, 20].

3.1.1 NS-2 Structure

NS-2 was written in C++ and OTcl (Tcl script language with Object-oriented extensions developed at MIT). In our simulation scenarios, the simulation codes are written in OTcl

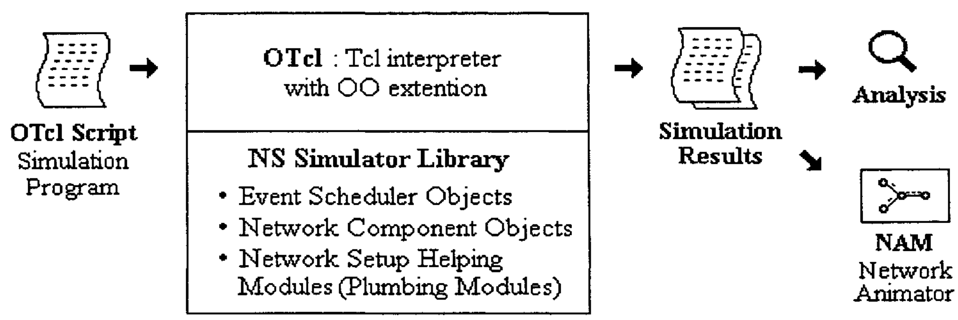

Figure 3.1 OTcl and C++ in NS-2

and models are written in C++ as shown in Figure 3.1

In our normal user's view, NS is Object-oriented Tcl (OTcl) script interpreter that has a simulation event scheduler and network component object libraries, and network setup module libraries. In other words, to use NS, we program in OTcl script language. To setup and run a simulation network, we should write an OTcl script that initiates an event scheduler, sets up the network topology using the network objects and the plumbing functions in the library, and tells traffic sources when to start and stop transmitting packets through the event scheduler. The term "plumbing" is used for a network setup, because setting up a network is plumbing possible data paths among network objects by setting the "neighbor" pointer of an object to the address of an appropriate object. When we want to make a new network object, we can easily make an object either by writing a new object or by making a compound object from the object library, and plumb the data path through the object. The power of NS comes from this plumbing.

NS is written not only in OTcl but in C++ also. For efficiency reason, NS separates the data path implementation from control path implementations. In order to reduce packet and event processing time (not simulation time), the event scheduler and the basic network component objects in the data path are written and compiled using C++. These compiled objects are made available to the OTcl interpreter through an OTcl linkage that creates a matching OTcl object for each of the C++ objects and makes the control functions and the configurable variables specified by the C++ object act as member functions and member variables of the corresponding OTcl object. In this way, the controls of the C++ objects are given to OTcl. It is also possible to add member functions and variables to a C++ linked OTcl object. The objects in C++ that do not need to be controlled in a simulation or internally used by another object do not need to be linked to OTcl. Likewise, an object (not in the data path) can be entirely implemented in OTcl. Figure 3.2 shows an object hierarchy example in C++ and OTcl. One thing to note in the figure is that for C++ objects that have an OTcl linkage forming a hierarchy, there is a matching OTcl object hierarchy very similar to that of C++.

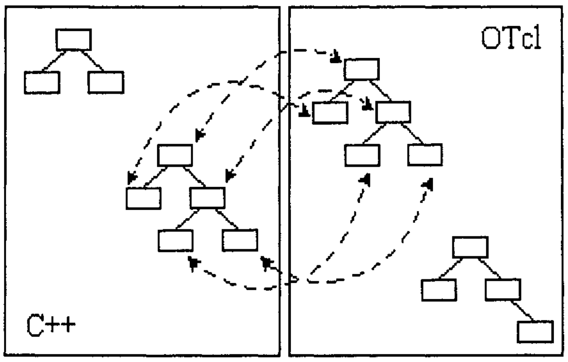

Figure 3.2. C++ and OTcl: The Duality

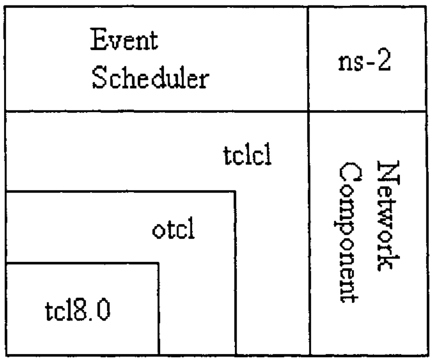

Figure 3.3 Architectural View of NS-2

Figure 3.3 shows the general architecture of NS. In this figure a general user (not an NS developer) can be thought of standing at the left bottom corner, designing and running simulations in Tcl using the simulator objects in the OTcl library. The event schedulers and most of the network components are implemented in C++ and available to OTcl through an OTcl linkage.

3.1.2 Network Components

A node is a compound object composed of a node entry object and classifiers as shown in Figure 3.4. A node has an address classifier that does routing and a port classifier.

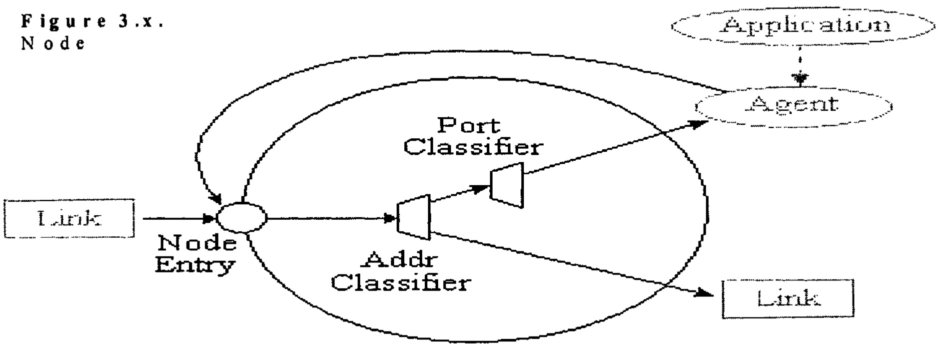

Figure 3.4 Node in NS-2

A link is another major compound object in NS. When a user creates a link using a duplex-link member function of a simulator object, two simplex links in both directions are created as shown in Figure 3.5.

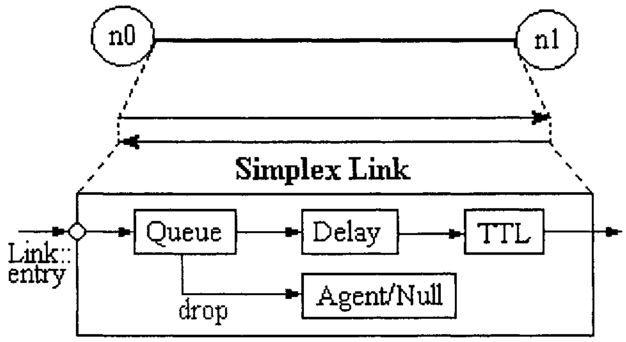

Figure 3.5 Link in NS-2

One thing to note is that an output queue of a node is actually implemented as a part of simplex link object. Packets dequeued from a queue are passed to the Delay object that simulates the link delay, and packets dropped at a queue are sent to a Null Agent and are freed there. Finally, the TTL object calculates Time To Live parameters for each packet received and updates the TTL field of the packet.

In NS, our simulation activities are traced around simplex links. If the simulator is directed to trace network activities (see sample Tcl codes in APPENDIX A & B: $ns trace-all file or $ns namtrace-all file), the links created after the command will have the following trace objects inserted as shown in Figure 3.6.

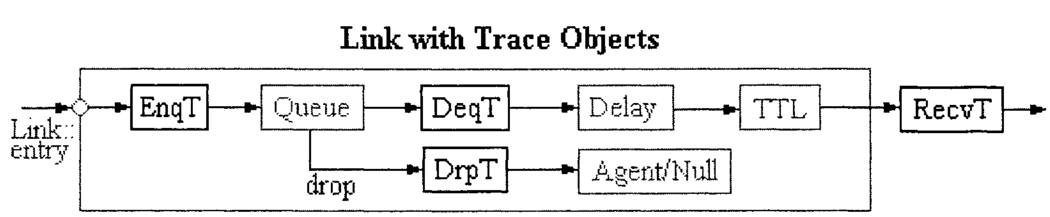

Figure 3.6 Inserting Trace Objects

When each inserted trace object (i.e. EnqT, DeqT, DrpT and RecvT) receives a packet, it writes to the specified trace file without consuming any simulation time, and passes the packet to the next network object.

The two most important network components are node and link. Figure 3.7 shows internals of an example in our simulation network setup and packet flow. The network consists of two nodes (n0 and n1) of which the network addresses are 0 and 1 respectively. A UDP

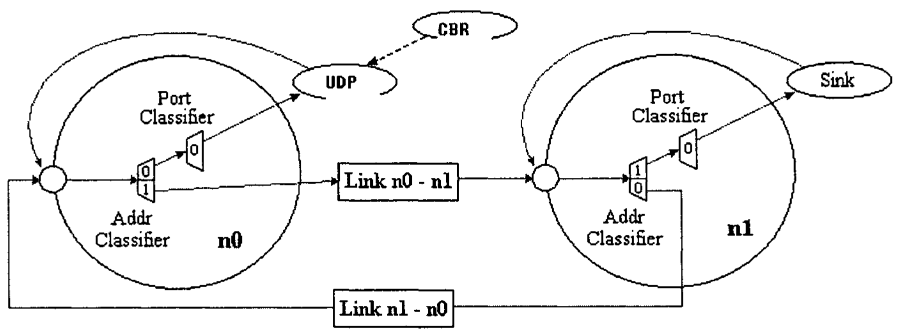

Figure 3.7 Packet Flow in NS-2

agent attached to n0 using port 0 communicates with a UDP sink object attached to n1 port 0. Finally, a CBR traffic source is attached to the UDP agent, asking to send some amount of data.

3.2 Modifications on NS-2

We made several modifications to monitor and optimize QoS parameters in the simulation environment. First, in order to monitor QoS in system models, we designed a delay performance module and a collision detection module. Second, in order to compare H-MANET architecture with traditional centralized control architecture, we integrated non-ad hoc routing protocols NOAH into NS-2. Third, in order to measure power consumption in H-MANET architecture, we designed a power consumption module based on transmission range. Finally, in order to test impact of mobility on H-MANET, we designed a mobility scenario generator.

3.2.1 Delay Monitor and Collision Counting

NS is basically an OTcl interpreter with network simulation object libraries. As we introduced in the last section, NS is implemented in both OTcl and C++. Extending NS by adding a new basic network object usually involves working around making OTcl linkage from C++ code, since the object class in the data path should be written in C++ for efficiency reasons. Here we present C++/OTcl linkages available in NS by adding new modules in C++.

The goal of delay monitor module is to record the end-to-end delays for each single packet, and then we can get an average number for each communication link which represents the delay performance in QoS. In delay monitor module, we recorded the time of the instances in the scheduler when a packet is initialized. At monitoring side, we stamp the time of the instance one more time, by which, we can obtain the time difference spent by this packet's travel. Meanwhile, we also need to create C++/OTcl linkages for new variables. The procedures and functions described here can be found modifications in *NS/common/packet.h, agent.cc, NS/tools/lossmonitors.cc,* and, *NS/tcl/lib/ns-default.tcl.* In our simulations, we will accumulate all the delays caused by different communication links and take an average to evaluate the delay performance of a system.

Basically, as we discussed in section 2.1.2, the protocol stack we use in our study model is an 802.11-based network. The delays in communication link are mainly caused by collisions [21]. CSMA/CA mechanism is very effective when the medium is not heavily loaded, since it allows stations to transmit with minimum delay. But there is always a chance of stations simultaneously sensing the medium as free and transmitting at the same time, causing a collision. These collision situations must be identified so the packets can be retransmitted by the MAC layer, rather than by the upper layer. The collisions cause the significant delay. According to the 802.11 standard, a maximum of seven retransmissions are allowed before the frame is dropped. We inserted new module at MAC layer to automatically count the number of collision in each node. Meanwhile, we also need to create C++/OTcl linkages for new variables. The procedures and functions described here can be found in modifications in *NS/MAC/mac-802_11.h, mac-802_11.cc* and *NS/common/packet.h.*

After the modification, the delay and collision counting modules are transparent to all users. You can find how these modules can be invoked in OTcl scripts.

3.2.2 Power Consumption

A wireless ad hoc network usually consists of mobile devices with limited battery power. Thus, energy-efficient communication techniques are very important. The most common technique is the power control scheme, in which a node transmits data packets to its neighbor at the minimum required power level [22].

In wireless networks, the power of a transmitted signal is attenuated at the rate of $1/d^n$, where d is the distance between the sender and receiver and n is the path loss exponent between 2 and 6. Accordingly, transmitting data packets directly to a node may consume more energy than going through some intermediate nodes. Based on this observation, most of the proposed energy-efficient routing protocols have tried to find a path that has many short-range hops in order to consume the least amount of total energy. Meanwhile, we are supposed to minimize the average hop number, in order to obtain delay and throughput performances.

As implemented in NS-2, energy module is a node attribute. The energy module represents level of energy in a mobile node. The energy module in a node has an initial value which is the level of energy the node has at the beginning of the simulation. This is known as *initialEnergy_*. It also has a given energy usage for every packet it transmits and receives. These are called *txPower_* and *rxPower_*. On the other hand, physical layer controls the transmission range of each mobile node, but energy module and transmission range are independent. Based on some experiences [23], we derived a formula that properly calculates energy consumption based on transmission range in an ad hoc networking environment. The procedures and functions described here can be found in modifications in *ns/energymodel[.cc and.h]*, and *ns/wireless-phy.cc.*

3.2.3 Mobility Generator

The mobility generator is used to generate mobility scenarios used to evaluate the QoS performance in MANET. NS provides a simple mobility generator for F-MANET; however, in our simulations, we need to simulate the H-MANET architecture, which needs a totally different mobility generator based on clusters.

In H-MANET, cluster mobility can be used in cluster-based communication. Here, each cluster has a logical center CH. Initially, each MN of the cluster is uniformly distributed in the neighborhood of the CH. Subsequently, at each instant, every node has a speed and direction that are derived randomly deviating from that of the CH. Inside each cluster, each MH moves in Random Waypoint Model, but when it can not move out of cluster boundary. We assume that there is no handoff in our simulations.

In our mobility generator, we input cluster number, the number of nodes in each cluster and locations of CHs; the output will be compatible with the requirements of NS mobility input standard. Basically, this portion of coding is independent C++ code, which is not integrated with NS, but provides NS input format.

3.3.4 NOAH

NOAH is a wireless routing agent that (in contrast to DSDV, DSR, ...) only supports direct communication between wireless nodes or between APs and MNs. This allows us to simulate scenarios where multi-hop wireless routing is undesired, for example, the WLAN and TDM system which are widely used in the market. NOAH does not send any routing related packets.

In order to integrate the NOAH routing protocol in NS-2, we integrate a new routing agent NOAH, and set up its support for hierarchical address. The procedures and functions are implemented in the following steps: 1) In *Makefile*, add *noah/noah.o* \ to *OBJ_CC* and *tcl/mobility/noah.tcl* \ to *NS_TCL_LIB* 2) add *noah.h* and *noah.cc* to a new subdirectory *noah/,* where *noah.h* and *noah.cc* are new files created to handle routing protocols. 3) add *noah.tcl* to *tcl/mobility/, noah.tcl* makes the routing in tcl codes possible. 4) modifications on *tcl/lib/ns-lib.tcl.h* to integrate the new routing agent.

In OTcl scripts, we can invoke the new routing protocol just like DSDV. The new routing agent will forward every packet to its CH and search for destination, even if the source and destination are located in the same cluster.

Chapter Summary

In this chapter, we introduced NS-2, a simulation tool to create system models and platform for simulations. We did some modifications on the platform to implement our H-MANET architecture. The insertion of the new modules is transparent to users, the invoking blocks can be found in APPENDIX A & B (two typical simulation cases for F-MANET and H-MANET respectively)

Chapter 4

Simulation Results and Discussion

We use simulation approach to investigate the H-MANET's performances, and compare those with the F-MANET and the WLAN. Result of this study will enable us to broaden our knowledge about hierarchical ad hoc architectures and give us information if hierarchy indeed provides performance improvements?

For performance analysis, we look into Quality of Service (QoS), which includes capacity performance, delay performance and power performance. As we discussed in the first chapter, the F-MANET suffers from poor scalability, therefore we will evaluate the scalability of the H-MANET and compare that with F-MANET.

4.1 Experiment Definition

We simulated system model *Static One-hop H-MANET* as we introduced in Chapter 2. This model is a simple model to represent the H-MANET architecture and the system is stationary. B.Liu and Z.Liu [8] consider the throughput capacity of hybrid one-hop wireless networks. Analytical expressions of the throughput capacity are obtained. For a hybrid of n node and m base stations, the results show that if m grows asymptotically faster than $\sqrt{n}$, the benefit of adding base stations on capacity is significant. The results enable us to know some basic throughput behaviors of Static One-hop H-MANET.

We also simulated H-MANETs with multi hops in wireless links and mobility in motions. The system models used in those experiments are also mentioned in the chapter 2, where we can systematically evaluate the more QoS parameters in more general models.

In order to compare the F-MANET and the H-MANET architectures, we defined some parameters which will be used in our discussions:

Throughput: It is measured as the total number of useful data (in bytes per second) received at the destinations, averaged over the steady state duration of a simulation. The network's total transmit capacity can be obtained by summing all throughputs of the source-destination pairs per second. The reported numbers here are the total network throughput averaged over the number of mobile nodes.

Delay: It is measured as the end-to-end delay (in seconds) for one link over the steady state duration of the simulation. The system's total delay can be obtained by summing all links' delays. The reported numbers here are the total system delay averaged over the received number of packets. In hierarchical architecture, reported delay also includes the delay caused by the wired links.

Delivery Ratio: It is measured as the successful number of received packet divided by sum of the number of dropped and received packets during the steady state section of the simulations.

Scalability: It is influence of increasing node densities for given area or increasing coverage area for given node densities on the QoS (Delay, throughput and power performances) of the network.

Energy Consumption: It is measured as the total amount of energy consumption for transmitting and sending packets at wireless interfaces of each node, averaged over the steady state duration of a simulation. The network's total energy consumption can be obtained by summing all energy consumptions in the system during the simulation process. Energy consumption in H-MANET networks will include wireless interfaces at CHs.

Power Efficiency: It is measured as the total amount of energy consumption divided by the total successfully received byte number (watt/byte). The reported number is collected during the steady state section of the simulations.

Mobility: The mobility scenarios are designed with different pause time. The mobility model is as follows: at every instance, a node randomly chooses a destination and moves towards it with a velocity chosen uniformly randomly form $[0, V]$, where V is the maximum allowable velocity for every mobile node. After reaching the destination, the node stops for a duration defined by the 'pause time' parameter. After this duration, it again chooses a random destination and repeats the whole process again until the simulation ends.

4.2 One-Hop System

In this study, H-MANET and F-MANET architectures use a static one-hop system model. Such static scenarios will be performance benchmark for the future scenarios where more hop number and impact of mobility will be investigated.

4.2.1 Traffic Analysis

We consider two kinds of ad hoc architectures: Hierarchical and Flat. Both of the architectures are deployed over comparable geographical areas, which is 2, 4 and 8 times of a 400 meters (m) by 400 meters square flat region. The transmission range of wireless interface is fixed at 250m (Note: A typical transmission range in WLAN, such as Lucent WaveLan WLAN Adapter). Each CH is located in the center of each cluster, which makes the one hop system possible.

The connection patterns and locations of the mobile nodes are randomly chosen for each simulation case. During the H-MANET simulations, all the inter-cluster traffic was channeled through upper tier nodes, which were connected via wired links, and the intra-cluster traffic was routed as in traditional ad hoc networks.

A series of simulation experiments for the two architectures were conducted using the system model and parameter outlined above. The key parameters are summarized in Table 4.1.

Table 4.1 Simulation Parameters of One-hop System

	H-MANET	F-MANET
Cluster Number	**2,4,8**	**N/A (Same Coverage Area)**
Coverage Area	**2,4,8* (400m*400m)**	**Same as H-MANET**
Routing Protocol	**DSDV**	**DSDV**
MAC Layer	**802.11b**	**802.11b**
PHY & Trans. Range	**2Mbps, 250m**	**2Mbps, 250m**
Packet Type and Size	**CBR, 512 Bytes**	**CBR, 512 Bytes**
Link(Node) Number	**80**	**80**
Mobility	**Static**	**Static**

We increase the input data little by little in these two systems. Packet generation rate is defined how many packets will be generated in one second per node.

A cluster in the H-MANET architecture covers an area of 400m by 400m and has one CH at the center of the square linked by wired links, where CHs are connected with 10 Mbps bandwidth and 1 millisecond (ms) propagation delay. For each H-MANET, there is a F-MANET having same number of nodes that covers same geographical area to have meaningful comparison of the two architectures. For example, If H-MANT has four clusters with 400m by 400m for each cluster. The counterpart F-MANET covers same geographic area, which is 800m by 800m, with same number of nodes.

Observations:

Both of the architectures are deployed over comparable geographical areas, which is 2, 4 and 8 times of a unit cluster. We denote three sets of systems: H2, F2; H4, F4; and H8, F8 based on different coverage areas. HX stands for hierarchical architecture with X clusters; and FX stands for flat architecture with same coverage areas of HX architecture. The throughput capacity performance and delay performance are shown in Figure 4.1 and 4.2 respectively.

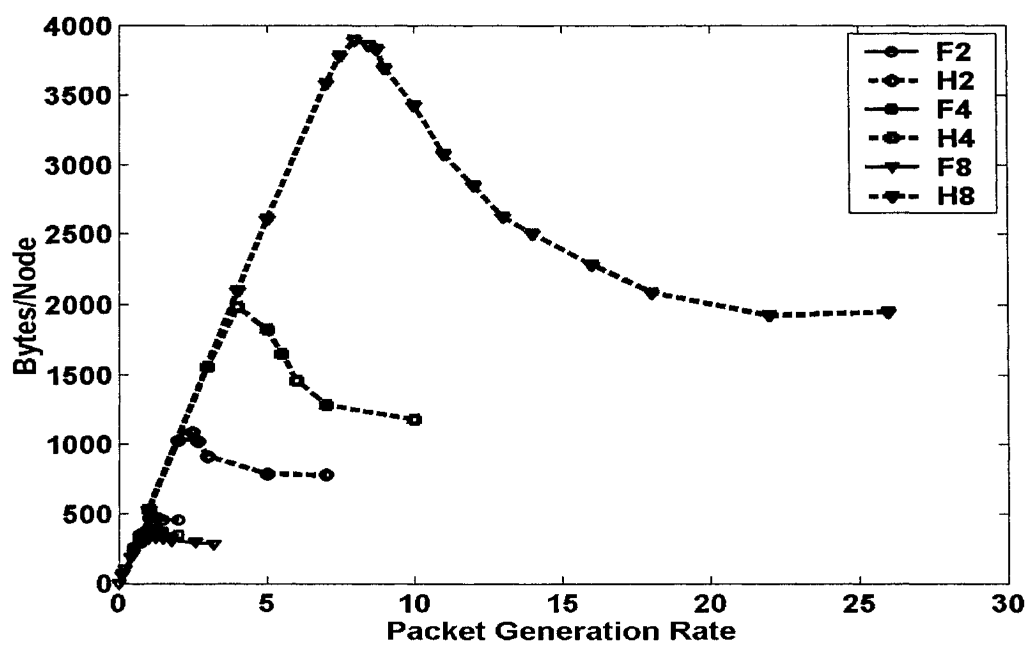

Figure 4.1 Throughput Performance of F-MANET and H-MANET

For both architectures, we tested the networks with same traffic loads. As we know in communication systems, the throughput is maximized when the maximum departure rate is reached. In Figure 4.1, the lines before the saturated points are the stable regions we are interested. We truncated the unstable regions where simulation tool is unable to correctly produce the results due to limited memory. We observed that for any offered load (packet generation rate) H-MANET network provides better throughput and lower delays than F-MANET network.

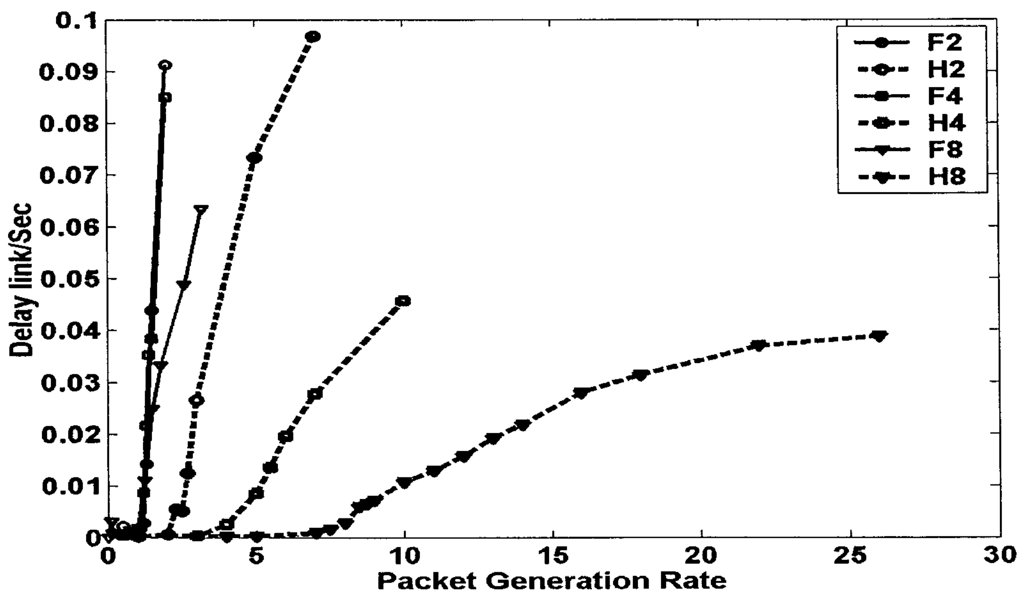

Figure 4.2 Delay Performance of F-MANET and H-MANET

Table 4.2 Throughput and Delay Improvements in H-MANET networks

Cluster Number	H-MANET Through-put B/Node	F-MANET Through-put B/Node	**Through-put Imp.**	H-MANET Delay Sec./Link	F-MANET Delay Sec./Link	**Delay Imp.**
2	1050	480	**2.19**	0.0052	0.002	**0.38**
4	2000	390	**5.15**	0.0025	0.005	**2.00**
8	3900	320	**12.19**	0.0028	0.01	**3.57**

Table 4.2 shows the Maximum throughputs in two architectures with different cluster numbers, where we can compare the hierarchical and flat networks and see how many times the performance will be improved. The delay values indicated in the table are the delays when the max throughputs are reached. For example, for 4-cluster simulation, hierarchical network is 5 times better in throughput performance and it is 2 times better in delay performance.

It seems that throughput performance improvements in H-MANET increase with the cluster number, the relationship between throughput performance and cluster number are plotted in Figure 4.3. We note that the throughput for each node increases linearly with cluster numbers, which means if you double your cluster number, your maximum throughput for each node will be doubled. Base on the above relationship, we can say that the cluster number can be adjusted to meet the throughput requirement of the network.

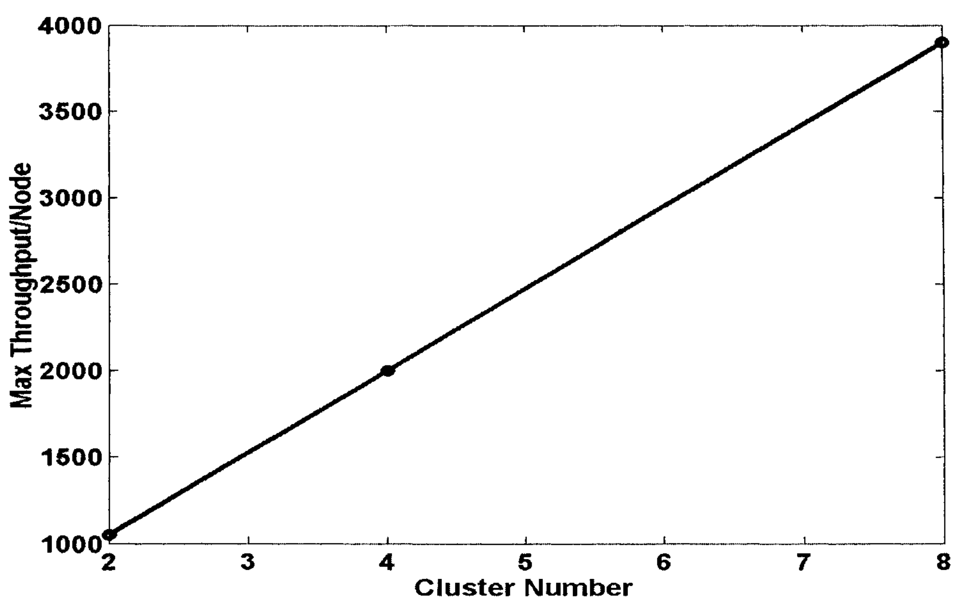

Figure 4.3 Reachable Throughput with Different Cluster Number

We also look into delivery ratio, which means what percent of packets will be successfully delivered to their destinations. We take the H8 and F8 networks as typical examples, their delivery ratios with different traffic load are plotted in Figure 4.4. From the figure, we note that the delivery ratios drop if the packet generation rates increase in both architectures; however, the H-MANET is able to successfully deliver much more packets than F-MANET does for any given delivery ratio; for example, H-MANET's throughput is approximately 16 times more than F-MANET when delivery ratio is 80% for two systems.

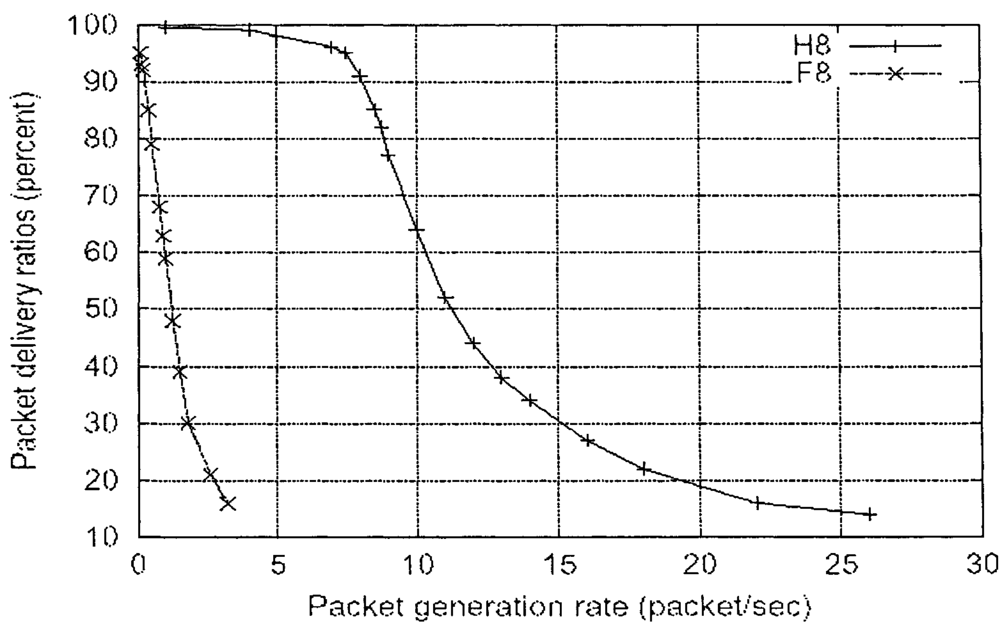

Figure 4.4 Delivery Ratio Vs Packet Generation Rate

Discussion:

The above simulations show that the throughput and delay performances of the F-MANET architecture are highly improved with the number of clusters. Those suggest that H-MANET networks provide wider trade-off margins for delay and throughput in desired packet transmission rates. A network can be designed to satisfy desired capacity and delay performance simultaneously.

We can also obtain an experimental formula about the relationship between cluster number and maximum obtainable capacity:

$$C(n,m) = \theta(mW)$$

Where, n is the MN number, m is the CH number and W is the transmission rate in the communication channels. However, we note that too many CHs in the network will introduce complexity in routing protocols and consume bandwidth to transmit management information, which will offset the benefits from clustering; therefore, setting a proper number of CH number will make the QoS maximized.

In the experiment, higher packet generation rates cause lower delivery ratios because offered load to network need to be compatible with the network throughputs. We note that for the same delivery ratio in Figure 4.4, the capacity improvement of the hierarchical network is significant; and influence of delivery ratios on throughput is much more visible than flat network. This suggests that by adjusting packet generation rates (offered load) at the nodes, the network capacity can be optimized more easily.

If we investigate the architecture of H-MANET, we note that the inter-cluster traffic will go through upper tier without broadcasting to all the MNs. Even with two tiers, H-MANET outperforms F-MANET significantly. The reasons can be attributed to two factors:

1) H-MANET architecture greatly reduced the number of hops that a packet travels

2) Having wired links (i.e. dedicated links) among CHs eliminate the wireless medium sharing for inter-cluster traffic, which enable the nodes to transmit simultaneously

Therefore, H-MANET networks can carry more data with smaller delays. Therefore, the most important parameters in QoS improved in H-MANET architecture.

4.2.2 Scalability

As we mentioned in section 1.3, traditional F-MANET networks have flat architecture, which suffers from poor scalability. Some theoretical analysis was presented in Section 2.3.

Gupta and Kumar [1] showed the relationship between capacity performance and node number in F-MANET architecture. However, we would like to know the relationship not only capacity performance, but also delay performance and node number as well. Based on the assumption in our system modes. We design the simulations as follows:

First of all, we need to define a new parameter "node density", which is number of nodes in one unit area. In our simulation, we control the cluster size to ensure each wireless link is able to reach CH in one hop, where we assume the cluster size is 400m*400m. Therefore, the unit of node density can also be defined node number per cluster in our simulation. In order to fairly compare two architectures, we change the node number and cluster number in two architectures, and investigate scalability of H-MANET architecture.

We consider two kinds of architectures: F-MANET and H-MANET, both of the architectures are deployed over comparable geographical areas, which is 2, 4 and 8 times of 400 meters (m) by 400 meters square flat region. The connection patterns and locations of the mobile nodes are randomly chosen for each simulation case. The key parameters are summarized in Table 4.3.

We increase the node numbers in these two systems. From Figure 4.4, we know that delivery ratio is adjustable by controlling packet generation rate. Therefore, we control the packet generation rate to make sure that the delivery ratios in all simulations are fixed to a reasonable value 80%, from which, we are able to investigate the scalability of two systems.

The same assumptions are used in our simulations. A cluster in the H-MANET architecture covers an area of 400m by 400m and has one CH at the center of the square linked by wires to other, where CHs are connected with 10 Mbps bandwidth and 1 millisecond (ms) propagation delay.

Table 4.3 Simulation Parameters of Network Scalability

	H-MANET	F-MANET
Cluster Number	**2,4,8**	**N/A(Same Coverage Area)**
Coverage Area	**2,4,8* (400m*400m)**	**Same as H-MANET**
Routing Protocol	**DSDV**	**DSDV**
MAC Layer	**802.11b**	**802.11b**
PHY & Trans. Range	**2Mbps, 250m**	**2Mbps, 250m**
Packet Type and Size	**CBR, 512 Bytes**	**CBR, 512 Bytes**
Link(Node) Number	**24,40,80,160**	**24,40,80,160**
Mobility	**Static**	**Static**
Delivery Ratio	**80%**	**80%**

Observations:

For each H-MANET, there is a F-MANET having same number of nodes that covers same geographical area to meaningfully compare two architectures. The capacity performance with different node numbers in two systems is plotted as in Figure4.5. The figure shows the obtainable throughput for each MN in two systems with increasing node numbers. The experimental results show that throughput for each node decreases rapidly toward zero while the number of nodes increases in flat architecture. And we can also see from the bottom three curves in Figure 4.5, which belongs to flat architecture, that, for the same number of nodes, the capacity decreases when coverage area increases. For better understanding of the relationships among number of nodes, capacity and delay, Figure 4.6 and 4.7 are plotted for different node densities, which are throughput and delay performance respectively.

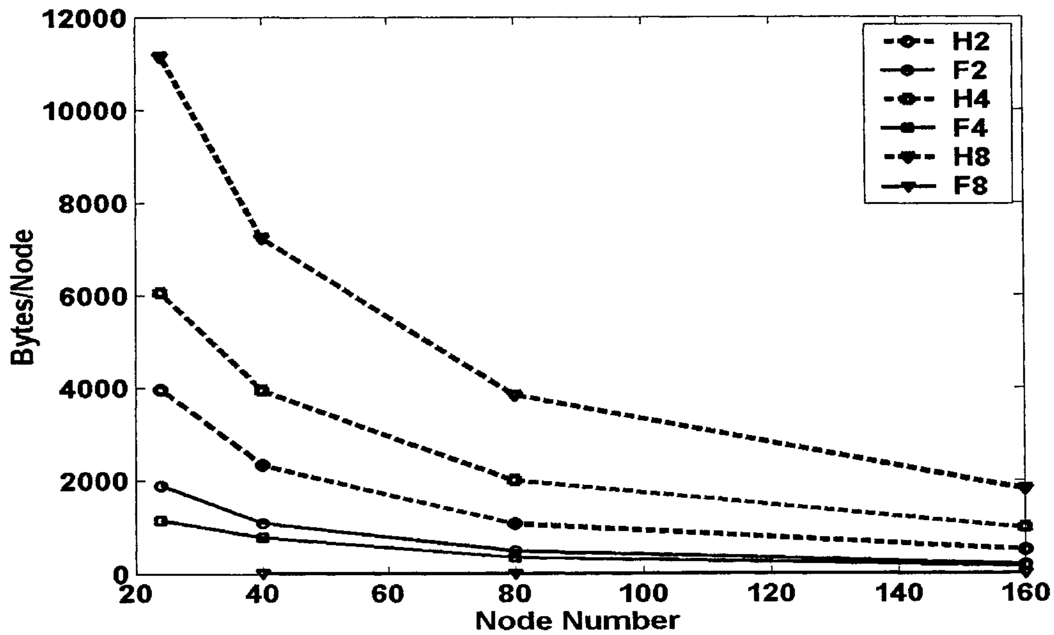

Figure 4.5 Throughput Vs Node Number (Delivery Ratio=80%)

Figure 4.6 does not produce new experimental data, we just reorganize the output presentation from node number to node density. The top three lines are H-MANET works and the bottom three lines are F-MANET networks.

For any given node densities, like 10, 20 or 40 nodes per cluster, the obtainable throughput by each node in the F-MANETs varies significantly; for example, when the node density is 10 nodes per cluster, the F-MANET with 8 cluster's throughput capacity is only approximately one tenth of the throughput in the F-MANET with 2 clusters. Therefore, a F-MANET network enlarges, but the throughput performance decreases quickly, which means F-MANETs does not scale. On the other hand, in the same node density simulation, H-MANET networks have very similar throughput performance around 4000bytes/second/node. When H-MANET network enlarges, the throughput performance degrades very little. And the throughput performance of H-MANET networks are always better than corresponding F-MANET networks.

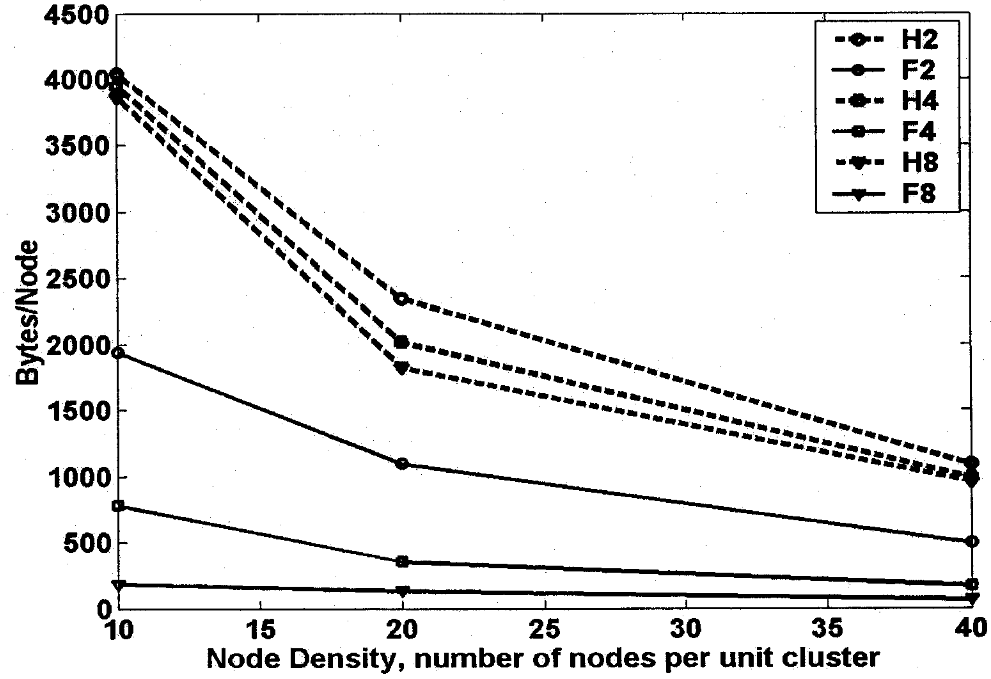

Figure. 4.6 Throughput Vs Node Density (Delivery Ratio=80%)

Figure 4.7 shows delay performance in two systems with different node densities. We note that, for average delay of each link, H-MANET networks get closer to F-MANET networks when network scale enlarges, especially when the system has high node density. For example, there is no visual difference between two systems when the node density is 40. One remark is that these delay were measured when the H-MANET networks have much higher throughput capacities.

Discussions:

We guaranteed that the delivery ratios in all the experiments are above 80%, which is a very typical scenario in real world. We have done some analysis based on the results form those simulations. Regarding the throughput, we note that, in F-MANET networks, if we fix node density and expand the network's geographic area, network capacity decreases quickly at any given node densities, which strongly suggests that large-scale flat architecture does not scale. Conversely, we note that H-MANET networks do not strongly depend on the number of clusters, or geographic area of the networks, which indicates that the scalability of the hierarchical architecture scales well even with two tiers. In our H-MANET networks, the wireless medium collisions are less likely to take place and the communications link is more likely to be successfully established. It is obvious that the impact of network scale on H-MANET throughput capacity is much less than F-MANET networks.

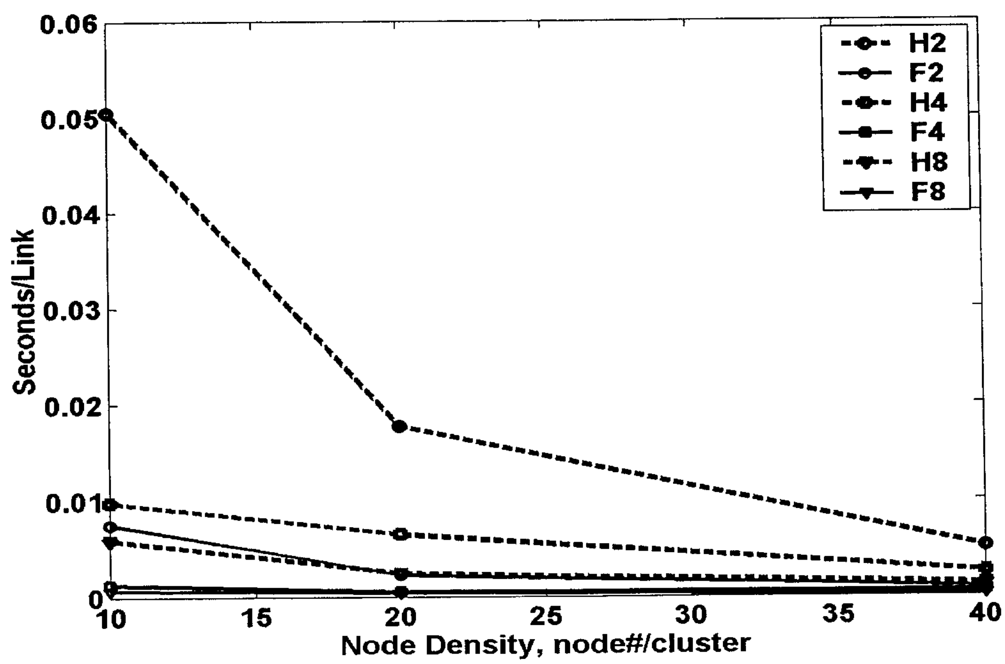

Figure. 4.7 Delay Vs Node Density (Delivery Ratio=80%)

Regarding the delay performance, as we observed from Figure 4.7, H-MANET networks will be becoming better and better if the network gets large. The reason behind the figure can be attributed to two factors:

1) For small networks, the average hop number in a F-MANET is small. Clustering scheme probably would increase the average hop number in the links (Minimum

hop number is three in H-MANET, but only one in F-MANET). Moreover, the upper tier network adds extra delays in the entire link (1ms in the simulations).

2) For large networks, the average hop number in a F-MANET is large. Clustering scheme will probably decrease the average hop number in the links. The upper tier's delay can be omitted in this case.

Based on the analysis, we conclude that clustering scheme in H-MANET always helps to eliminate wireless sharing, which is always good for improving throughput capacity. On the other hand, cluster scheme decreases increasing trend of the average hop number, which will greatly decrease delay when the network scale gets larger.

The results that we obtained from our simulations clearly demonstrate that the tradeoff between capacity and delay is very clear; for example, if we need to put stricter constraints for delay, we can trade off certain throughput performance to meet the goal. Such results strongly suggest that QoS provisioning is more flexible in H-MANET architecture than F-MANET. By adjusting number of clusters and number nodes per cluster, desired delay, throughput and delivery ratios of the network can be provided for wide range of offered loads. Such optimization or trade offs are not available in flat architecture.

4.3 Multi-Hop Systems

In this study, the Static Multi-hop H-MANET, Mobile Multi-hop H-MANET and WLAN system models will be investigated and compared. Multi-hop H-MANET is a more general system model in our study objective and we will look into more parameters, such as power efficiency and mobility.

In this section's simulations, a network with 4 clusters is used as an H-MANET profile to be evaluated. The transmission range of transceivers is adjustable to control hop numbers inside the clusters.

4.3.1 Increasing Hop Number

We adjust the transmission range radii of CHs and MNs between 200m to 400m. Because the cluster disk's radius is 400m, the hop number will be ranging from 1 hop to 2 hops. Basically, if the transmission range increases, the average hop number for each link decreases. However, greater transmission introduces more interference to be involved. The key parameters are summarized in Table 4.4.

Table 4.4 Simulation Parameters of Multi-hop Systems

	H-MANET	F-MANET	WLAN
Cluster Number	**4**	**N/A**	**4**
Coverage Area	**4* Disks (Radius =400m)**	**Disk (Radius=800m)**	**4* Disks (Radius =400m)**
Routing Protocol	**DSDV**	**DSDV**	**NOAH**
MAC Layer	**802.11b**	**802.11b**	**802.11b**
Trans. Range	**200-400**	**200-400**	**400**
PHY Layer	**2Mbps**	**2Mbps**	**2Mbps**
Packet Type and Size	**CBR, 512 Bytes**	**CBR,512 Bytes**	**CBR,512 Bytes**
Node Number	**80**	**80**	**80**
Link Number	**40**	**40**	**40**
Mobility	**Static**	**Static**	**Static**
Packet Generation Rate	**5 Pkt/S**	**5 Pkt/S**	**5 Pkt/S**

As we know, the area of a disk follows the formula $A=\pi r^2$, where r is the radius of a disk. The F-MANET network area $A=\pi r^2=\pi *(800)^2 =\pi *4*(400)^2=4* \pi *(400)^2$; therefore, these three system models have same coverage area. Routing protocol NOAH is used in WLAN system model as described in section 2.2.3, which is a non-ad hoc routing protocol. In order to ensure the connectivities in WLAN, we fix the transmission range in the WLAN system model to be 400m, which makes it comparable with the other two system models.

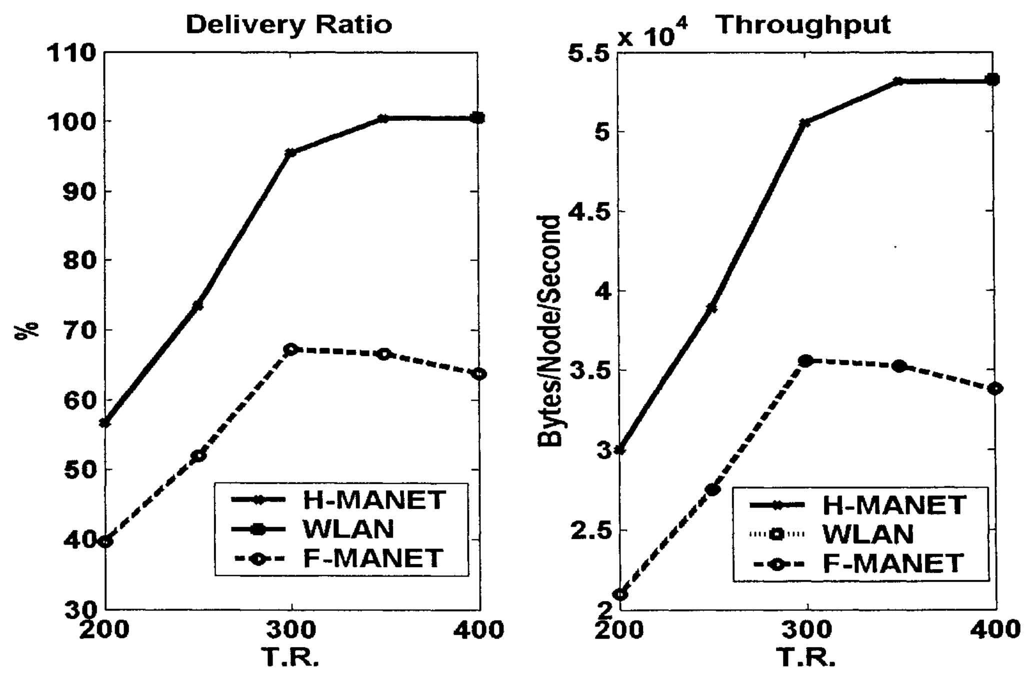

Figure 4.8 Multi-hop Throughput and Delivery Ratio

Observations:

We have same amount of input for three system models and they have same connection pattern. The throughput performance and delivery ratios for these three system models are shown in Figure 4.8. All the successfully delivered packets are counted both in throughput performance and delivery ratio performance. From the figure, we observe that H-MANET network has much better throughput capacity over F-MANET network in all transmission ranges, or in all average hop numbers. The two-tier infrastructure architecture approximately doubles the throughput capacity of F-MANET network in our simulations.

Delivery ratio can easily reach over 90% in H-MANET network if we have transmission range over 300m. For F-MANET network, transmission range will not keep improving throughput capacity because interference offsets benefits from the reducing of average hop number. As we can see in F-MANET networks, throughput performance drops when transmission range is over 300m in our simulations. From this figure, we know that the clustering scheme plays a significant role in eliminating interferences and improving throughput capacities. Even in H-MANET networks, reducing the average hop number (or increasing the transmission range) can significantly improve the throughput capacities. The throughput performance of WLAN in our simulation is the same as H-MANET, because both of them can reach 100% delivery ratio when transmission range is maximized.

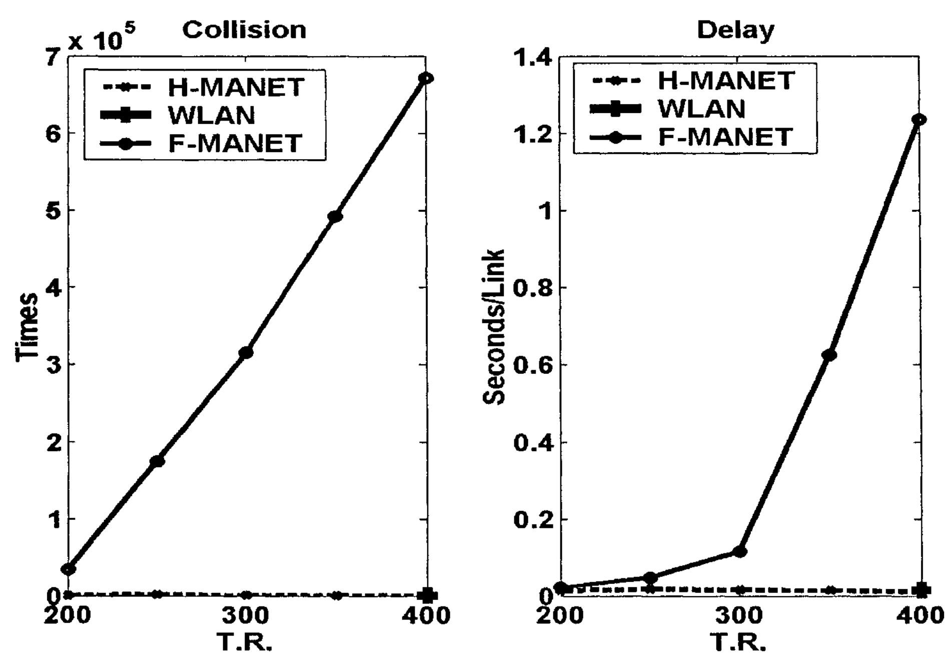

Figure 4.9 Multi-hop Collision and Delay

In order to further investigate the interference in the above three system models, we inserted collision monitoring, where we are able to count how many times of total collisions in the different system models. In the protocol of 802.11b, a maximum of seven retransmissions are allowed before the packet is dropped, which determine the busy medium in wireless channel is the main cause for degrading delay performance.

The monitored collisions and delay performance for three system models are shown in Figure 4.9, where collision number is the number captured in the entire simulation process and delay is the average number for each link. The WLAN networks are measured only in the maximum transmission range (400m).

In F-MANET networks, we observed that number of collisions keep increasing when the transmission range increases, which means larger transmission range of a MN causes larger interference in the MNs around itself. By using a cluster scheme in the F-MANET network, the collision number is dramatically reduced.

As we can observe in Figure 4.9, delay keeps increasing along with increasing collision numbers in F-MANET networks. Compared with F-MANET systems, F-MANET networks and WLAN networks have very good delay performances. The delay will not vary significantly in different transmission range (or hop number) by using clustering schemes. A similar result will be observed in collision numbers. H-MANET and WLAN systems have similar delay performance and both of them significantly overpass the performance in F-MANET system.

Energy consumption is a very important parameter in wireless QoS evaluation, which determines the lifetime of a MN and determines the lifetime of the whole system. Power efficiency is also a very important factor in evaluation system's performance. Higher power efficiency will enable system to transmit and receive more packets for the same energy consumption. The data of energy consumption collected in our simulation is measured on one MN in one second, and the power efficiency is measured as " how many bytes can be successfully transmitted in one watt energy consumption. The result is shown in Figure 4.10.

The energy consumption of F-MANET networks keep increasing with the greater transmission range and its values are always larger than corresponding H-MANET networks. The value of energy consumption in H-MANET networks increases when throughput capacity increases (Transmission range increases). WLAN network's consumes a little bit more energy than H-MANET in maximum transmission range. For the power efficiency in Figure 4.10, we observe that F-MANET networks' power efficiency keep decreasing with larger transmission ranges, which means one watt energy sends less and less packets with larger transmission range in F-MANET networks. However, H-MANET has very good consistency in power efficiency performance. The hop number will not significantly vary the power performance. The WLAN system has power efficiency between H-MANET and F-MANET, but closer to H-MANET as the square symbol indicated in Figure 4.10.

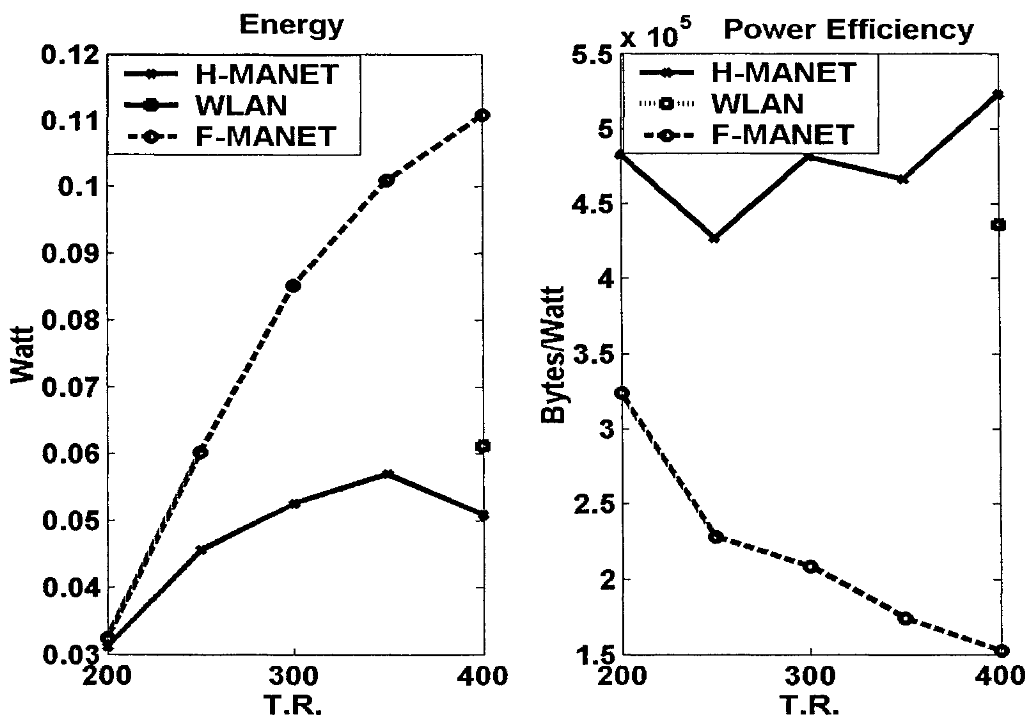

Figure 4.10 Multi-hop Energy Consumption and Power Efficiency

Discussion:

A static multi-hop H-MANET was analyzed in our simulation, and we compared the key QoS parameters with F-MANET and WLAN systems.

There is an optimal value in F-MANET's transmission range to reach maximum throughput capacity. Too large transmission range will result in small hop numbers in each link, and result in severe interferences on transmission link as well. Smaller transmission range results in the opposite effect, but cluster schemes can solve the contradiction efficiently. Throughput capacity is greatly improved by applying a clustering scheme to F-MANET system, which was demonstrated in our simulations. Delivery ratio can easily reach 100% with infrastructure support. At this point, H-MANET and WLAN systems have similar performances on throughput.

The delays in F-MANET system will be extremely large if interference in the network is severe. The main reason for causing large delay is the continuous collisions in the wireless channel and the larger transmission range results in more collisions. The cluster scheme is able to significantly eliminate the collisions at the MAC layer. Compared with F-MANET, H-MANET and WLAN systems keep collision number at a very low level. The different wireless channels are used in different clusters, which avoids the competition in the wireless medium in different clusters. The WLAN system has slightly less delay compared with the H-MANET system, because H-MANET networks have more hop numbers in each communication link.

F-MANET systems consume most energy and transmits least throughput in three systems. The power efficiency in F-MANET keeps decreasing with the increasing transmission range. H-MANET systems have steady performance in power efficiency and they exceed WLAN system in power efficiency. In H-MANET systems, if the source and destination in a link are located within one cluster, it will be routed without a CH being involved. However, in WLAN system, all the communication traffic has to go through AP, which is the dominanted reason of more energy consumed.

Basically, the QoS improvement in multi-hop H-MANET is obvious in terms of throughput, delay and power. Setting proper transmission range for hop number control is a very effective way to meet the required QoS parameters.

4.3.2 Mobility Analysis

The system model of the *Mobile Multi-hop H-MANET* was introduced in Section 2.2.5, where mobility scenarios with different pause times are employed. The moving speed of MNs are chosen ranging between 0 and 60km/hr. We control all the MNs moving inside its own cluster boundary.

Table 4.5 Simulation parameters of Mobility

	H-MANET	F-MANET
Cluster Number	4	N/A
Coverage Area	**4* Disks (Radius =400m)**	**1* Disks (Radius =800m)**
Routing Protocol	**DSDV**	**DSDV**
MAC Layer	**802.11b**	**802.11b**
PHY & Trans. Range	**2Mbps, 250m**	**2Mbps, 250m**
Packet Type and Size	**CBR, 512 Bytes**	**CBR, 512 Bytes**
Node Number	**80**	**80**
Link Number	**40**	**40**
Mobility	**Mobile**	**Mobile**
Speed of MN	**0-60km/hr**	**0-60km/hr**
Packet Generation Rate	**5 Pkt/S**	**5 Pkt/S**

In this section's simulations, we try to investigate the mobility's impact on H-MANET. The motivation of simulations is to investigate whether the properties of H-MANET we obtained from previous investigation would vary or not. The key parameters are summarized in Table 4.5.

This simulation model is exactly the same 4-disk H-MANET network as we used in the last section. The only change is that waypoint mobility scenarios are introduced in system analysis.

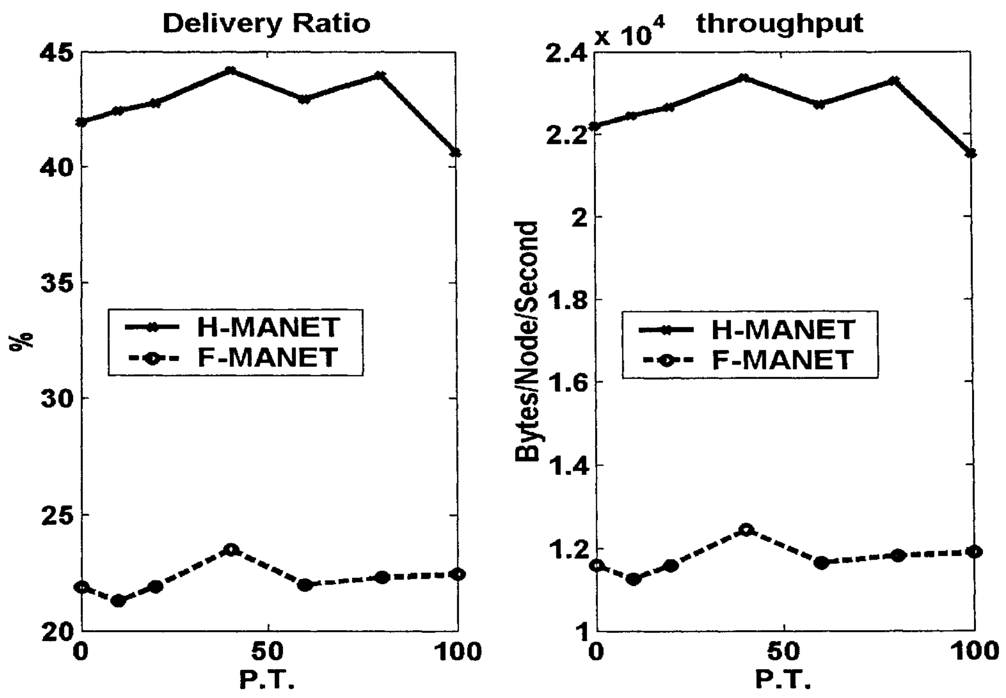

Figure 4.11 Delivery Ratio and Throughput in Mobility

<u>Observations:</u>

In our simulations, nodes are mobile and system topology changes very frequently. The transmission range of 250m is defined in the simulations; therefore, all MNs are able to reach its CH in one or two hops. When the pause time increases, the system becomes less and less dynamic. The system is a static system if we set the pause time to be infinity. We set up the pause time ranging from 0 to 100 seconds. We investigate QoS parameters as we did in the previous simulations.

Figure 4.11 shows the delivery ratios and throughput performance in two different architectures, F-MANET and H-MANET. As we observed, the throughput capacity of F-MANET drops slightly if the system becomes more dynamic. In Figure 4.11 we observe that the H-MANET system always delivers more packets than F-MANET system. In our simulation, 4-custer schemes are applied to F-MANET. On average, the obtainable throughput for each MN approximately doubled. The observed throughput performance in mobility scenarios is similar to the previous static system models.

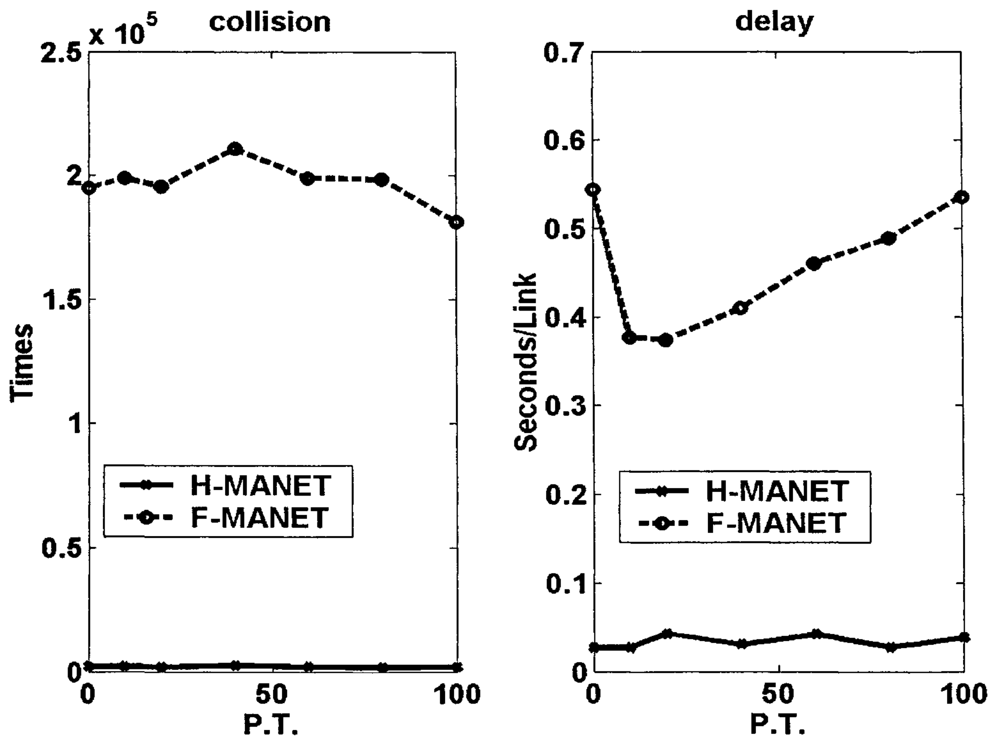

Figure 4.12 Collision and Delay in Mobility

Another important QoS parameter is delay performance. The frequently changing network topology breaks communication links easily. However, a cluster scheme makes the network less dynamic, because we do not consider handoffs in our simulations. Nodes are restricted in their own cluster by mobility generator and no extra overheads are necessary for handoffs. The observation is shown in Figure 4.12. H-MANET system seems to have more steady performance in delay. Less collision number is observed in H-MANET system as well. The properties are very similar to the previous observations.

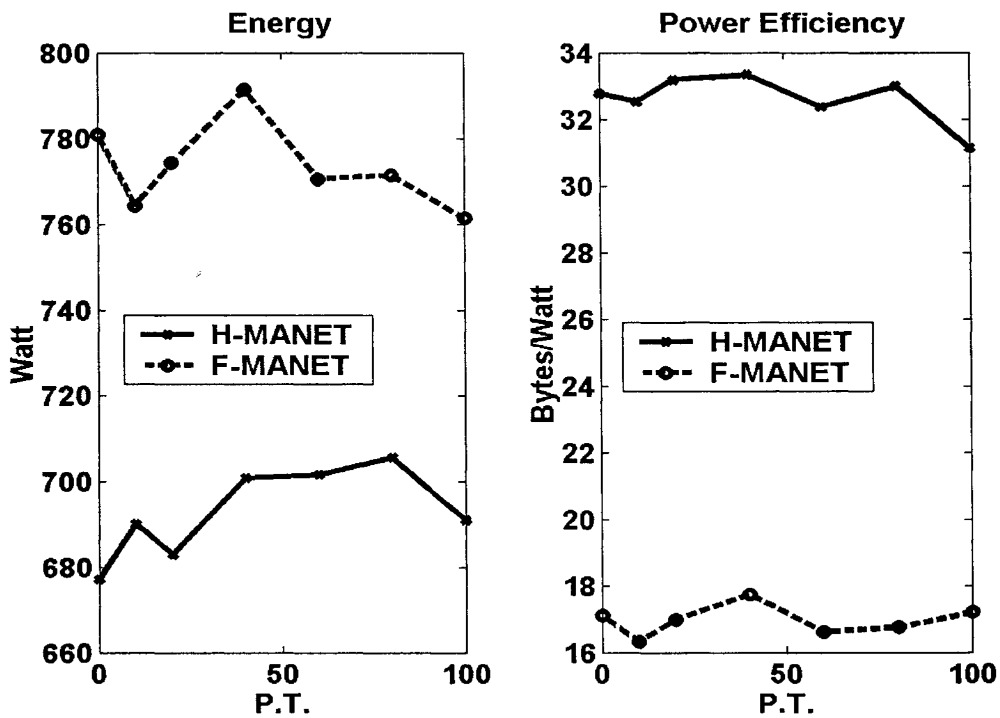

Figure 4.13 Energy Consumption and Power Efficiency in Mobility

Energy and power efficiency are observed in Figure 4.13. The energy consumption in H-MANET system includes the energy consumed at 4 CHs' wireless interfaces. Even through, we note that the H-MANET is almost doubled the power efficiency of F-MANET, which means that H-MANET network can send double packet with per unit energy consumed. The properties of pervious observations are reserved in varying pause times.

<u>Discussion:</u>

F-MANET network's throughput drops slightly, because frequently changing communication links produce more overhead than static. This is also the reason that F-

MANET has a slight variation in delay performance and energy consumption. By applying a cluster scheme, the network becomes less dynamic and system has more steady performance in all QoS parameters. All the QoS parameters of H-MANET system outperform the F-MANET system and have consistently good performance in different mobility scenarios.

Chapter Summary:

In this chapter, we examined the QoS of H-MANET architecture and compared it with two other architectures: unorganized architecture F-MANET, and centralized control architecture WLAN. Four typical system models were used to assist evaluation. Based on the simulations, we clearly understand how to improve F-MANET architecture's QoS in terms of throughput, delay and power efficiency, and how much clustering scheme is able to improve

However, we also need to mention that new node type called Cluster Head (CH) has to be introduced in the system., and some control management will be involved as well. Improperly designed system might offset all the benefits from clustering schemes.

Chapter 5

Conclusions

In this thesis, we have studied the QoS of Hierarchical ad hoc networks (H-MANET) in term of throughput, delay and power. The simulations have shown that there certainly is a need for H-MANET systems instead of F-MANET systems when the network scales.

5.1 Results

The key aspect when evaluating H-MANET architecture is to test them in realistic scenarios. We have tested it in different types of system models. H-MANET networks always outperform F-MANET in QoS performances, which clearly demonstrate the efficiency of clustering schemes. Moreover, H-MANET architecture has exhibited a good overall performance also when mobility is high. We observed the impact of hop number on F-MANET architecture and compared the performance with WLAN, where we found that H-MANET exhibited better power efficiency. We also observed the trade offs among QoS parameters in H-MANET networks. By carefully designing a hierarchical network, for example, properly setting up cluster number and hop number in cluster scheme, the network can meet required QoS parameters.

Compared with traditional ad hoc networks, the key conclusions of H-MANET architectures from simulations can be summarized as follows:

- Capacity and delay improvements
- Less impact on QoS from mobility
- Less power consumption and higher power efficiency
- More clusters, better performance.
- Trade-off between QoS parameters available

How much improvement we will obtain depends on the clustering schemes; for example, how many clusters, how large is one cluster and how many nodes in one cluster etc.

The results presented in this paper will be benchmark for those future studies. To my knowledge, this thesis is the first work which systematically evaluates the performance of hierarchical ad hoc networks.

5.2 Further Studies

Ad-hoc network is a rather hot concept in wireless communications. This means that there is much research going on and many issues that remain to be solved. Due to limited time, we have only focused on the most important QoS parameters of H-MANET under some ideal assumptions. However there are many issues that could be subject to further studies.

First of all, the simulator environment could be improved. These are just some of the improvements that could be made:

- Handoff schemes should be considered in the future, which is necessary in real networks
- Currently, CHs have one wireless interface and one wired interface respectively. Both of them should be wireless interfaces in the future, which is more practical to compare performance with traditional ad hoc networks.

Secondly, many issues related to hierarchical ad-hoc networks could be improved.

- Protocol stack needs some modifications. At network layer, DSDV is a protocol supporting hierarchical routing but its performance drops quickly in high-speed mobility. At MAC layer, 802.11 was specifically designed for WLAN, which produces unfairness problem in ad hoc system. At PHY layer, if more wireless interfaces were applied, channel distribution would be considered.

- More complicated mathematical models should be established to assist systematic analysis, although this is very difficult in the wireless world.

References

[1] Gupta, P.; Kumar, P.R.; "The capacity of wireless networks", *IEEE Transactions on Information Theory*, Volume: 46 , Issue: 2 , March 2000 Pages:388 – 404

[2] Bansal, N.; Liu, Z.; "Capacity, delay and mobility in wireless ad-hoc networks" , *Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies*, Volume: 2 , 30 March-3 April 2003 Pages:1553 - 1563

[3] Toh, C.-K.; Delwar, M.; Allen, D.; "Evaluating the communication performance of an ad hoc wireless network" *IEEE Transactions on Wireless Communications*,, Volume: 1 , Issue: 3 , July 2002 Pages:402 – 414

[4] M.J. Neely and E. Modiano; "Capacity and Delay Tradeoffs for Ad-Hoc Mobile Networks" *First International Conference on Broadband Networks, 2004*., 25-29 October. 2004 Pages:428 - 438

[5] Comaniciu, C. and Vincent Poor, l; "On the Capacity of Mobile Ad hoc Networks with Delay Constraint" in *Proceedings of the IEEE Infocom*, Jun 2002, Volume. 3, Pages. 1577--1586.

[6] Wan, PJ;Calinescu, G.; Yi, CW; "Minimum-power multicast routing in static ad hoc wireless networks" *IEEE/ACM Transactions on Networking,* Volume: 12 , Issue: 3 , June 2004 Pages:507 – 514

[7] Ramanathan, R. and Rosales-Hain, R.; "Topology control of multi-hop wireless networks using transmit power adjustment" in *IEEE INFOCOM,* Tel Aviv, Israel, March 2000. Pages:404 – 413

[8] Liu, B.; Liu, Z.; Towsley, D.; "On the capacity of hybrid wireless networks" INFOCOM 2003. *Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies*. IEEE , Volume: 2 , 30 March-3 April 2003 Pages:1543 - 1552

[9] Kozat, U. and Leandros; "Throughput Capacity of Random Ad Hoc Networks with Infrastructure Support", *Proceedings of the 9th annual international conference on Mobile computing and networking* MobiCom 2003 San Diego, CA, USA, Pages: 55 - 65

[10] Perkins, C. and Bhagwat, P.; "DSDV Routing over a Multi-hop Wireless Networks", *In Proceedings of INFOCOM'97,* Kobe, Japan, April 1997 Pages: 754 - 761

[11] "Wirless LAN Medium Access Control and Physical Layer Specifications" *IEEE Computer Society LAN MAN Standards Committee.* New York, 1997. IEEE Std. 802.11-1997

[12] Li, J.; Jannotti, J.; De Couto; Karger, D. R.; Morris, R.; "A Scalable Location Service for Geographic Ad Hoc Routing", *Proceedings of the 6th ACM International Conference on Mobile Computing and Networking,* Boston, Massachusetts, August 2000, Pages 120-130

[13] li, J.; Blake, C; "Capacity of Ad-hoc wireless Networks" *In the Proceedings of the 7th ACM International Conference on Mobile Computing and Networking*, Rome, Italy, July 2001. Pages: 61 - 69

[14] P. Roshan, J. Leary; " 802.11 Wireless LAN Fundamentals" Published by Cisco Press, 2003, ISBN: 1-58705-077-3

[15] Netzwerke, C. Bettstetter , C. Wagner, "The Spatial Node Distribution of the Random Waypoint Mobility Model", *Mobile Ad-Hoc 1. deutscher Workshop über Mobile Ad-Hoc Netzwerke* WMAN 2002, March 25-26, 2002, Pages 41-58,

[16] Resta, G.; Santi, P; "An analysis of the node spatial distribution of the random waypoint mobility model for ad hoc networks", *Proceedings of the second ACM international workshop on Principles of mobile computing*, October 2002, Toulouse, France Pages 44-50

[17] Falll, K and Varadhan, K.; "Ns Notes and Documentation Technical report" UC Berkery, LBL,USC/ISI, and Xerox PARC 2003

[18] Lucio, G. F.; Paredes-Farrera, M.; Jammeh E.; Fleury, M. and Reed, M. J.; "OPNET Modeler and Ns-2: Comparing the Accuracy of Network Simulators for Packet-Level Analysis using a Network Testbed" in the *WSEAS Transactions on Computers*, No.3, Volume. 2, July 2003, ISSN 1109-2750 , Pages 700-707.

[19] Jae Chung and Mark Claypool "NS by Example" http://nile.wpi.edu/NS/

[20] "NS tutorial" http://www.isi.edu/nsnam/ns/tutorial/, Marc Greis's tutorial for the UCB/LBNL/VINT Network Simulator "ns"

[21] Xu, S.; Saadawi, T.; "Does the IEEE 802.11 MAC protocol work well in multihop wireless ad hoc networks?" *Communications Magazine, IEEE* , Volume: 39 , Issue: 6 , June 2001 Pages 130 - 137

[22] Jung, E. and Vaidya, N. H., "A power control MAC protocol for ad hoc networks," in the *proceedings of MOBICOM'02*, September 2002. Atlanta, Georgia, Pages. 36-47

[23] Feeney, L.M.; Nilsson, M.; "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment" *Proceedings of Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies*, Volume: 3, 22-26 April 2001 Anchorage, Alaska, Pages:1548 - 1557

APPENDIX A

F-MANAT TCL SIMULATION CODE

```
#==========================================================
#F-MANET.tcl
#Simulation of a flat area (a disk with radius=800m) scenario
#Designed by Tai Yu. 2003-3004
#==========================================================
# # Define options
#
# channel type
set val(chan)  Channel/WirelessChannel;
# radio-propagation model
set val(prop)   Propagation/TwoRayGround;
# network interface type
set val(netif) Phy/WirelessPhy;
# MAC type
set val(mac) Mac/802_11;
# interface queue type
set val(ifq)  Queue/DropTail/PriQueue;
# link layer type
set val(ll) LL  ;
# antenna model
set val(ant) Antenna/OmniAntenna;
# max packet in ifq
set val(ifqlen) 50;
# number of mobilenodes
set val(nn) 40;
# routing protocol
set val(adhocRouting) DSDV;

set val(energymodel)     "EnergyModel"
set val(initenergy)    9999.0


set val(cp)     "./nn40-mc20-r5"
set val(sc)     "./f_40_p100"
# x coordinate of topology
set val(x)     1600;
# y coordinate of topology
set val(y)     1600;

# seed for random number gen.
set val(seed)  0.9;
# time to stop simulation
set val(stop)  200 ;

set val(sink)  20
# Transmission Range
```

```
set val(TR)    250;


# # check for boundary parameters and random seed
if { $val(x) == 0 || $val(y) == 0 } {
  puts "No X-Y boundary values given for wireless topology\n"
}
if {$val(seed) > 0} {
  puts "Seeding Random number generator with $val(seed)\n"
  ns-random $val(seed)
}

# create simulator instance
set ns_   [new Simulator]

# Create topography object
set topo   [new Topography]


set tracefd  [open tra.tr w]
set namtrace [open ani.nam w]
$ns_ trace-all $tracefd
$ns_ namtrace-all-wireless $namtrace $val(x) $val(y)

# define topology
$topo load_flatgrid $val(x) $val(y)

# create God
set god_ [create-god $val(nn)]

# node cofiguration
$ns_ node-config -adhocRouting $val(adhocRouting) \
             -llType $val(ll) \
             -macType $val(mac) \
             -ifqType $val(ifq) \
             -ifqLen $val(ifqlen) \
             -antType $val(ant) \
             -propType $val(prop) \
             -phyType $val(netif) \
             -channelType $val(chan) \
                          -topoInstance $topo \
             -wiredRouting OFF\
             -agentTrace ON \
             -routerTrace OFF \
             -macTrace OFF \
                            -energyModel $val(energymodel) \
                            -rxPower 0.3 \
                            -txpower 0.6\
                            -initialEnergy $val(initenergy)

#setting new Pt_

set RCV_th  [Phy/WirelessPhy set RXThresh_]
set new_pt [expr $RCV_th*pow($val(TR),4)/pow(1.5,4)]
```

```
puts "new Pt_ is $new_pt"

#Phy/WirelessPhy set Pt_ 0.2818
Phy/WirelessPhy set Pt_ $new_pt


for {set i 0} {$i < $val(nn) } {incr i} {
  set node_($i) [$ns_ node]
  $node_($i) random-motion 0    ;# disable random motion
}

# source connection-pattern and node-movement scripts
if { $val(cp) == "" } {
  puts "*** NOTE: no connection pattern specified."
      set val(cp) "none"
} else {
  puts "Loading connection pattern..."
  source $val(cp)
}

for {set i 0} {$i < $val(nn)} {incr i} {
   $ns_ initial_node_pos $node_($i) 20
}

if { $val(sc) == "" } {
  puts "*** NOTE: no scenario file specified."
      set val(sc) "none"
} else {
  puts "Loading scenario file..."
  source $val(sc)
  puts "Load complete..."
}

# Define initial node position in nam

$ns_ at [expr $val(stop)+0.0000001] "count_col";
#------------------
set mn_tol_col 0
set mn_tol_en 0


proc count_col { } {            ;#-count collision number--------------

  global val node_ mn_tol_col mn_tol_en

  for {set i 0} {$i < $val(nn) } {incr i} {
  set MYMAC_($i) [$node_($i) set mac_(0)]                ;
#------------------
      set colcount [$MYMAC_($i) set tai_collision_count_]
#      puts "MH $i 's collision count is $colcount"
#puts "Id of node_($i) is [$node_($i) id]"
set MYENG_($i) [$node_($i) energy]
      puts "Remaining energy of node_($i) is $MYENG_($i)"
      set mn_tol_en [expr ($mn_tol_en+$MYENG_($i))]
```

```
 set mn_tol_col [expr ($mn_tol_col+$colcount)]

  }
       puts "Total collision number of MNs is $mn_tol_col"
  set mn_tol_en [expr ($val(nn)*$val(initenergy)-$mn_tol_en)]
  puts "Total total power consumption of MNs is $mn_tol_en "

}
# Tell all nodes when the simulation ends
for {set i } {$i < $val(nn) } {incr i} {
   $ns_ at $val(stop).001 "$node_($i) reset";
}

#file to put throughput values
#
set f0 [open ./out.tr w]
set count 0
set bytes_num 0
set delay_num 0
set lost_num 0
set rvd_num 0
proc record { } {

global sink_  f0 val  count bytes_num delay_num lost_num rvd_num ;#node_
  set ns [Simulator instance]
  set time 0.5
      #set time 2.0
  set bw 0
      set delay 0
      set npckt 0
      set nls 0
      set tmp3 0
  set tmp 0
      set tmp1 0
  set tmp2 0
  set now [$ns now]

  for {set i 0} {$i <  $val(sink)} {incr i} {

  set tmp [$sink_($i) set bytes_]
      set bw [expr $bw+$tmp]
  $sink_($i) set bytes_ 0
      set tmp1 [$sink_($i)  set ketdelay_ ]
      set delay [expr $delay+$tmp1]
      $sink_($i)  set ketdelay_ 0
      set tmp2 [$sink_($i) set npkts_]
  set npckt [expr $npckt+$tmp2]
  $sink_($i)  set npkts_  0

  set tmp3 [$sink_($i) set nlost_]
  set nls [expr $nls+$tmp3]
  $sink_($i)  set nlost_  0

       }
```

```
#----------------------------------------------------
      while {$now >=[expr $time*50]} {
      puts $f0 "$now      [expr $bw/$time]   [expr $delay/($npckt+0.0001)] "
  puts "$now     [expr $bw/$time]   [expr $delay/($npckt+0.0001)] "

  break;
      }

  if {$now>=100} {
     set count [expr $count+1]
  set bytes_num [expr $bytes_num+$bw/$time]
  set delay_num [expr $delay_num+$delay/($npckt+0.0001)]
      set lost_num [expr $nls+$lost_num]
  set rvd_num [expr $rvd_num+$npckt]
      }
  #set MYMAC0 [$node_(0) set mac_(0)];
#-----------------
      #set colcount0 [$MYMAC0 set tai_collision_count]
      #puts "node_(0)'s collision count is $colcount0 now"

  $ns at [expr $now+$time] "record"
   }

$ns_ at 0.0 "record"
$ns_ at $val(stop).0003 "puts \"NS EXITING...\" ; $ns_ halt"
$ns_ at $val(stop).0002 "stop"
proc stop {} {
   global ns_ tracefd namtrace f0 count bytes_num delay_num lost_num rvd_num mn_tol_col mn_tol_en
val
#   $ns_ flush-trace
   close $tracefd
   close $namtrace
   close $f0

   puts "the average bytes is:"
   puts "[expr $bytes_num/$count]"
   puts "the average delay is:"
   puts "[expr $delay_num/$count]"
   puts "the received packets number"
   puts "$rvd_num"
   puts "the dropt packets number"
   puts "$lost_num"
   puts "ratio=[expr $rvd_num*1.0/($rvd_num+$lost_num)]"
   puts "ratio2=[expr $rvd_num*1.0/(5.0*20*100)]"
   puts "total collisions number is "
   puts "$mn_tol_col"
   puts "total power consumption is "
   puts "$mn_tol_en"
   puts Flat_$val(adhocRouting)=\[[format "%d %.1f %.0f %.4f %.3f %d %0.2f" $val(TR) $val(seed) [expr
$bytes_num/$count] [expr $delay_num/$count] \
   [expr $rvd_num*1.0/(5.0*20*100)] $mn_tol_col mn_tol_en]\]
   exec xgraph out.tr -geometry 800x400 &
```

```
   exit 0
}

# informative headers for CMUTracefile
puts $tracefd "M 0.0 nn $val(nn) x $val(x) y $val(y) rp \
  $val(adhocRouting)"
#puts $tracefd "M 0.0 sc $opt(sc) cp $val(cp) seed $valt(seed)"
puts $tracefd "M 0.0 prop $val(prop) ant $val(ant)"

puts "Starting Simulation..."
$ns_ run
```

APPENDIX B

H-MANAT TCL SIMULATION CODE

```
#======================================================
#H-MANET.tcl
#Simulation of a H-MANET system model with 4 disk area (a disk #radius=400m) scenario
#Designed by Tai Yu, 2003-2004
#======================================================
# Define options
#
# channel type
set val(chan)  Channel/WirelessChannel;
# radio-propagation model
set val(prop)  Propagation/TwoRayGround;
# network interface type
set val(netif) Phy/WirelessPhy;
# MAC type
set val(mac) Mac/802_11;
# interface queue type
set val(ifq)  Queue/DropTail/PriQueue;
# link layer type
set val(ll) LL  ;
# antenna model
set val(ant) Antenna/OmniAntenna;
# max packet in ifq
set val(ifqlen) 50;
# number of mobilenodes
set val(nn) 40;
# routing protocol
set val(adhocRouting) DSDV;

set val(energymodel)    "EnergyModel"
set val(initenergy)   9999.0

set opt(cp)      "./nn40-mc20-r5"
set opt(sc)      "./h4_40_p80"
# x coordinate of topology
set opt(x)     2400;
# y coordinate of topology
set opt(y)     2400;
# seed for random number gen.
set opt(seed)  0.3;
# time to stop simulation
set opt(stop)  200;
#$opt(nn)
set opt(sink)  20;
# Transmission Range
set opt(TR)    250;
```

```
set num_bs_nodes        4

# ================================================
# check for boundary parameters and random seed
if { $opt(x) == 0 || $opt(y) == 0 } {
  puts "No X-Y boundary values given for wireless topology\n"
}
if {$opt(seed) > 0} {
  puts "Seeding Random number generator with $opt(seed)\n"
  ns-random $opt(seed)
}

# create simulator instance
set ns_   [new Simulator]

# set up for hierarchical routing
$ns_ node-config -addressType hierarchical
AddrParams set domain_num_ 1          ;# number of domains
lappend cluster_num 4             ;# number of clusters in each domain
AddrParams set cluster_num_ $cluster_num
lappend eilastlevel  [expr $opt(nn)/4] [expr $opt(nn)/4] [expr $opt(nn)/4] [expr $opt(nn)/4] ;# number of
nodes in each cluster
AddrParams set nodes_num_ $eilastlevel ;# of each domain

set tracefd  [open wl4-out.tr w]
set namtrace [open wl4-60.nam w]
$ns_ trace-all $tracefd
$ns_ namtrace-all-wireless $namtrace $opt(x) $opt(y)

# Create topography object
set topo   [new Topography]

# define topology
$topo load_flatgrid $opt(x) $opt(y)

# create God
create-god [expr $opt(nn) + $num_bs_nodes]

$ns_ node-config -adhocRouting $opt(adhocRouting) \
            -llType $opt(ll) \
            -macType $opt(mac) \
            -ifqType $opt(ifq) \
            -ifqLen $opt(ifqlen) \
            -antType $opt(ant) \
            -propType $opt(prop) \
            -phyType $opt(netif) \
            -channelType $opt(chan) \
                    -topoInstance $topo \
            -wiredRouting ON \
                    -agentTrace ON \
            -routerTrace OFF \
            -macTrace OFF  \
                    -energyModel $opt(energymodel) \
                     -initialEnergy $opt(initenergy)
```

```
   #-rxPower 0.3 \
   #-txpower 0.4

#setting new Pt_

set RCV_th  [Phy/WirelessPhy set RXThresh_]
set new_pt [expr $RCV_th*pow($opt(TR),4)/pow(1.5,4)]
puts "new Pt_ is $new_pt"

#Phy/WirelessPhy set Pt_ 0.2818
Phy/WirelessPhy set Pt_ $new_pt

#create base-station node

set temp0(0) 0.0.0
for {set j 1} {$j<= [expr $opt(nn)/4]} {incr j} {
   set temp0($j)  0.0.$j
}

set temp1(0) 0.1.0
for {set j 1} {$j<= [expr $opt(nn)/4]} {incr j} {
   set temp1($j)  0.1.$j
}

set temp2(0) 0.2.0
for {set j 1} {$j<= [expr $opt(nn)/4]} {incr j} {
   set temp2($j)  0.2.$j
}

set temp3(0) 0.3.0
for {set j 1} {$j<= [expr $opt(nn)/4]} {incr j} {
   set temp3($j)  0.3.$j
}

set BS(0) [$ns_ node  $temp0(0)]
set BS(1) [$ns_ node  $temp1(0)]
set BS(2) [$ns_ node  $temp2(0)]
set BS(3) [$ns_ node  $temp3(0)]

$BS(0) random-motion 0
$BS(1) random-motion 0
$BS(2) random-motion 0
$BS(3) random-motion 0


$BS(0) set X_ 400
$BS(0) set Y_ 400
$BS(0) set Z_ 0

$BS(1) set X_ 400
$BS(1) set Y_ 2000
$BS(1) set Z_ 0

$BS(2) set X_ 2000
```

```
$BS(2) set Y_ 2000
$BS(2) set Z_ 0

$BS(3) set X_ 2000
$BS(3) set Y_ 400
$BS(3) set Z_ 0

$ns_ duplex-link $BS(0) $BS(1) 100Mb 1ms DropTail
$ns_ duplex-link $BS(1) $BS(2) 100Mb 1ms DropTail
$ns_ duplex-link $BS(2) $BS(3) 100Mb 1ms DropTail
$ns_ duplex-link $BS(3) $BS(0) 100Mb 1ms DropTail

# create mobilenodes in the same domain as BS(0)
# note the position and movement of mobilenodes is as defined
# in $opt(sc)

#configure for mobilenodes
$ns_ node-config -wiredRouting OFF

for {set j 0} {$j < [expr $opt(nn)/4]} {incr j} {
   set node_($j) [ $ns_ node $temp0([expr $j+1]) ]
   $node_($j) base-station [AddrParams addr2id [$BS(0) node-addr]]
}

for {set j [expr $opt(nn)/4]} {$j < [expr $opt(nn)/2]} {incr j} {
   set node_($j) [ $ns_ node $temp1([expr $j+1-$opt(nn)/4]) ]
   $node_($j) base-station [AddrParams addr2id [$BS(1) node-addr]]
}

for {set j [expr $opt(nn)/2]} {$j < [expr $opt(nn)/4*3]} {incr j} {
   set node_($j) [ $ns_ node $temp2([expr $j+1-$opt(nn)/2]) ]
   $node_($j) base-station [AddrParams addr2id [$BS(2) node-addr]]
}

for {set j [expr $opt(nn)/4*3]} {$j < $opt(nn)} {incr j} {
   set node_($j) [ $ns_ node $temp3([expr $j+1-$opt(nn)/4*3]) ]
   $node_($j) base-station [AddrParams addr2id [$BS(3) node-addr]]
}

# source connection-pattern and node-movement scripts
if { $opt(cp) == "" } {
  puts "*** NOTE: no connection pattern specified."
      set opt(cp) "none"
} else {
  puts "Loading connection pattern..."
  source $opt(cp)
}
if { $opt(sc) == "" } {
  puts "*** NOTE: no scenario file specified."
      set opt(sc) "none"
} else {
  puts "Loading scenario file..."
  source $opt(sc)
  puts "Load complete..."
```

```
}

# Define initial node position in nam

for {set i 0} {$i < $opt(nn)} {incr i} {

   # 20 defines the node size in nam, must adjust it according to your
   # scenario
   # The function must be called after mobility model is defined?????????
   $ns_ initial_node_pos $node_($i) 20

}

# Tell all nodes when the simulation ends
for {set i } {$i < $opt(nn) } {incr i} {
   $ns_ at $opt(stop).0004 "$node_($i) reset";
}
$ns_ at $opt(stop).000005 "$BS(0) reset";
$ns_ at $opt(stop).00003 "$BS(1) reset";
$ns_ at $opt(stop).00006 "$BS(2) reset";

$ns_ at $opt(stop).0001 "$BS(3) reset";




$ns_ at [expr $opt(stop)+0.000002] "count_col";
#------------------
$ns_ at [expr $opt(stop)+0.000001] "BS_count_col" ;
#------------------
set mn_tol_col 0
set bs_tol_col 0
set mn_tol_en 0
set bs_tol_en 0

proc count_col { } {;
#-count collision number----------------------------
  global opt node_ mn_tol_col mn_tol_en

  for {set i 0} {$i < $opt(nn) } {incr i} {
  set MYMAC_($i) [$node_($i) set mac_(0)]  ;
#------------------

      set colcount [$MYMAC_($i) set tai_collision_count_]
#       puts "MH $i 's collision count is $colcount"
  #puts "Id of node_($i) is [$node_($i) id]"

  set MYENG_($i) [$node_($i) energy]

      #puts "energy of node_($i) is $MYENG_($i)"
      set mn_tol_en [expr ($mn_tol_en+$MYENG_($i))]
  set mn_tol_col [expr ($mn_tol_col+$colcount)]
```

```
  }
      puts "Total collision number of MNs is $mn_tol_col"
   set mn_tol_en [expr ($opt(nn)*$opt(initenergy)-$mn_tol_en)]
   puts "Total total power consumption of MNs is $mn_tol_en "

}
#-count collision number-----------------------------
proc BS_count_col { } {;

global BS opt num_bs_nodes bs_tol_col bs_tol_en num_bs_nodes

  for {set i 0} {$i < $num_bs_nodes } {incr i} {
  set MYMAC_BS_($i) [$BS($i) set mac_(0)];
 #-----------------
  #set MYMAC_BS1_($i) [$BS($i) set mac_(1)];
 #-----------------
  set BS_colcount [$MYMAC_BS_($i) set tai_collision_count_]
  #puts "Id of BS_($i) is [$BS($i) id]"
  set MYENG_BS_($i) [$BS($i) energy]
 #puts "energy of BS_($i) is $MYENG_BS_($i)"
 # puts "BS $i 's collision count is $BS_colcount"



 set bs_tol_en [expr ($bs_tol_en+$MYENG_BS_($i))]
  set bs_tol_col [expr ($bs_tol_col+$BS_colcount)]
  }
 puts "Total collision number of BSs is $bs_tol_col"
  set bs_tol_en [expr ($num_bs_nodes*$opt(initenergy)-$bs_tol_en)]
  puts "Total total power consumption of BSs is $bs_tol_en "
}

#file to put throughput values
#
set f0 [open wl4out.tr w]
set count 0
set bytes_num 0
set delay_num 0
set lost_num 0
set rvd_num 0
proc record { } {

global sink_ f0 opt count bytes_num delay_num lost_num rvd_num
  set ns [Simulator instance]
  set time 0.5
  set bw 0
      set delay 0
      set npckt 0
  set tmp 0
      set tmp1 0
  set tmp2 0
  set tmp3 0
      set nls 0
  set now [$ns now]
```

```
  for {set i 0} {$i < $opt(sink)} {incr i} {

  set tmp [$sink_($i) set bytes_]
      set bw [expr $bw+$tmp]
  $sink_($i) set bytes_ 0
      set tmp1 [$sink_($i) set ketdelay_ ]
      set delay [expr $delay+$tmp1]
      $sink_($i) set ketdelay_ 0

      set tmp2 [$sink_($i) set npkts_]
  set npckt [expr $npckt+$tmp2]
  $sink_($i) set npkts_ 0

 set tmp3 [$sink_($i) set nlost_]
  set nls [expr $nls+$tmp3]
  $sink_($i) set nlost_ 0
     }



#----------------------------------------------------
      while {$now >=[expr $time*50]} {
      puts $f0 "$now        [expr $bw/$time]   [expr $delay/($npckt+0.001)] "
  puts "$now  [expr $bw/$time]   [expr $delay/($npckt+0.001)]"
  break;
      }
if {$now>=100} {
  set count [expr $count+1]
  set bytes_num [expr $bytes_num+$bw/$time]
  set delay_num [expr $delay_num+$delay/($npckt+0.001)]
  set lost_num [expr $nls+$lost_num]
  set rvd_num [expr $rvd_num+$npckt]
      }
  $ns at [expr $now+$time] "record"
}
$ns_ at 0.0 "record"
$ns_ at $opt(stop).0003 "puts \"NS EXITING...\" ; $ns_ halt"
$ns_ at $opt(stop).0002 "stop"
proc stop { } {
global ns_ tracefd namtrace f0 count bytes_num delay_num \
   lost_num rvd_num mn_tol_col bs_tol_col mn_tol_en bs_tol_en opt
#   $ns_ flush-trace
   close $tracefd
   close $namtrace
   close $f0

   puts "the average bytes is:"
   puts "[expr $bytes_num/$count]"
   puts "the average delay is:"
   puts "[expr $delay_num/$count]"
   puts "the received packets number"
   puts "$rvd_num"
   puts "the dropt packets number"
```

```
   puts "$lost_num"
   puts "ratio=[expr $rvd_num*1.0/($rvd_num+$lost_num)]"
   puts "ratio2=[expr $rvd_num*1.0/(5.0*20*100)]"
   puts "total collisions number is "
   puts "[expr ($mn_tol_col+$bs_tol_col)]"
   puts "total power consumption is "
   puts "[expr ($mn_tol_en+$bs_tol_en)]"

puts $opt(adhocRouting)=\[[format "%d %.1f %.0f %.4f %.3f %d %0.2f" $opt(TR) $opt(seed) [expr
$bytes_num/$count] [expr $delay_num/$count] \
   [expr $rvd_num*1.0/(5.0*20*100)] [expr ($mn_tol_col+$bs_tol_col)] [expr ($mn_tol_en+$bs_tol_en)]
]\]
   exec xgraph wl4out.tr  -geometry 800x400 &

   exit 0
}
# informative headers for CMUTracefile
puts $tracefd "M 0.0 nn $opt(nn) x $opt(x) y $opt(y) rp \
  $opt(adhocRouting)"
#puts $tracefd "M 0.0 sc $opt(sc) cp $opt(cp) seed $opt(seed)"
puts $tracefd "M 0.0 prop $opt(prop) ant $opt(ant)"

puts "Starting Simulation..."
$ns_ run
```

APPENDIX C

SIMULATION DATA

Note:

1) In the multi-hop systems, data are collected in three system models, which are static multi-hop H-MANET, F-MANET and WLAN

2) Acronyms: S.M. = System Model, Pro.=Routing Protocol, T.R. = Transmission Range, Thr. = Throughput (Bytes/Second/Node), Del. = Delay (Second/Link), D.R. = Delivery Ratio (*100%), Col. = Collision Number (Times), S.E.C. = System Energy Consumption(Watts)

3) Seed numbers are used to produce random number in creating connection pattern or node locations.

S.M.	Pro.	T.R.	Seed	Thr.	Del.	D.R.	Col.	S.E.C.
Hier.	DSDV	200	0.9	28956	0.0135	0.547	178	464.50
Hier.	DSDV	200	0.6	26436	0.0159	0.499	709	443.20
Hier.	DSDV	200	0.3	34588	0.0133	0.653	316	584.26
Flat	DSDV	200	0.3	15394	0.0144	0.291	14052	281.50
Flat	DSDV	200	0.6	24615	0.0223	0.465	57673	629.03
Flat	DSDV	200	0.9	22979	0.0276	0.434	30944	646.54
Hier.	DSDV	250	0.3	26251	0.0206	0.496	4217	608.66
Hier.	DSDV	250	0.6	47838	0.0177	0.904	545	890.26
Hier.	DSDV	250	0.9	42544	0.0154	0.804	387	687.11
Flat	DSDV	250	0.9	26522	0.0412	0.501	64450	876.94
Flat	DSDV	250	0.6	23826	0.0673	0.450	205412	936.89
Flat	DSDV	250	0.3	32190	0.0343	0.608	256208	1077.68
Hier.	DSDV	300	0.6	50453	0.0185	0.953	806	975.88
Hier.	DSDV	300	0.3	53290	0.0119	1.007	86	727.39
Hier.	DSDV	300	0.9	47901	0.0156	0.905	550	817.49
Flat	DSDV	300	0.3	33599	0.0420	0.635	283604	1138.99
Flat	DSDV	300	0.6	39115	0.2357	0.739	325936	1589.88
Flat	DSDV	300	0.9	34080	0.0710	0.644	336127	1362.64

S.M.	Pro.	T.R.	Seed	Thr.	Del.	D.R.	Col.	S.E.C.
Hier.	DSDV	350	0.6	53369	0.0164	1.008	340	969.30
Hier.	DSDV	350	0.3	53047	0.0160	1.002	374	921.05
Hier.	DSDV	350	0.9	53015	0.0134	1.002	193	843.74
Flat	DSDV	350	0.3	33143	0.6217	0.626	343012	1571.76
Flat	DSDV	350	0.6	35705	0.6915	0.674	657687	1610.18
Flat	DSDV	350	0.9	36922	0.5614	0.698	476104	1668.78
Hier.	DSDV	400	0.9	53290	0.0115	1.007	81	813.37
Hier.	DSDV	400	0.6	53094	0.0115	1.003	53	812.84
Hier.	DSDV	400	0.3	53025	0.0116	1.002	21	813.26
Flat	DSDV	400	0.3	31873	0.8817	0.602	892288	1693.21
Flat	DSDV	400	0.6	35811	1.4753	0.676	462941	1880.91
Flat	DSDV	400	0.9	33725	1.3514	0.637	659245	1747.25
WLAN	NOAH	400	0.3	53348	0.0147	1.008	339	979.13
WLAN	NOAH	400	0.6	53274	0.0146	1.006	179	979.61
WLAN	NOAH	400	0.9	53099	0.0147	1.003	169	975.80

APPENDIX D

MOBILITY SIMULATION DATA

Note:

1) In the mobility scenarios, data are collected in two system models, which are mobile multi-hop H-MANET and mobile F-MANET networks.

2) Acronyms: S.M. = System Model, P.T.=Pause Time (Seconds), T.R. = Transmission Range, Thr. = Throughput (Bytes/Second/Node), Del. = Delay (Second/Link), D.R. = Delivery Ratio (*100%), Col. = Collision Number (Times), S.E.C. = System Energy Consumption (Watts)

3) Seed numbers are used to produce random number in creating connection patterns, node locations and mobility scenarios.

S.M.	P.T	T.R.	Seed	Thr.	Del.	D.R.	Col.	S.E.C
Hier.	0	250	0.3	23711	0.0271	0.448	1856	688.19
Hier.	0	250	0.6	20640	0.0277	0.390	2153	672.92
Hier.	0	250	0.9	22240	0.0258	0.420	2362	670.34
Flat	0	250	0.3	10998	0.5594	0.208	183633	773.66
Flat	0	250	0.6	12980	0.5192	0.245	227273	806.73
Flat	0	250	0.9	10789	0.5544	0.204	173913	762.55
Hier.	10	250	0.3	24383	0.0254	0.461	2607	702.70
Hier.	10	250	0.6	22340	0.0284	0.422	2062	696.61
Hier.	10	250	0.9	20635	0.0275	0.390	1937	670.83
Flat	10	250	0.9	12732	0.4028	0.240	218328	789.10
Flat	10	250	0.6	10466	0.3040	0.198	182244	756.15
Flat	10	250	0.3	10622	0.4234	0.201	196382	747.91
Hier.	20	250	0.3	25432	0.0630	0.480	2347	697.29
Hier.	20	250	0.6	19078	0.0246	0.360	1545	646.22
Hier.	20	250	0.9	23468	0.0395	0.443	1901	705.40
Flat	20	250	0.3	11780	0.2846	0.223	187437	757.25
Flat	20	250	0.6	11832	0.3282	0.224	208179	803.63
Flat	20	250	0.9	11196	0.5090	0.211	191243	762.51

S.M.	P.T	T.R.	Seed	Thr.	Del.	D.R.	Col.	S.E.C
Hier.	40	250	0.9	23161	0.0283	0.438	2325	698.96
Hier.	40	250	0.6	23456	0.0330	0.443	2643	708.25
Hier.	40	250	0.3	23479	0.0301	0.444	2696	695.25
Flat	40	250	0.9	11906	0.4288	0.225	220936	798.15
Flat	40	250	0.6	13648	0.4114	0.258	208534	795.05
Flat	40	250	0.3	11801	0.3893	0.223	202924	781.04
Hier.	60	250	0.3	21043	0.0287	0.398	1669	695.98
Hier.	60	250	0.6	22976	0.0702	0.434	2487	710.56
Hier.	60	250	0.9	24128	0.0269	0.456	1946	698.20
Flat	60	250	0.3	10979	0.5343	0.207	209888	774.96
Flat	60	250	0.6	11650	0.4141	0.220	189973	754.34
Flat	60	250	0.9	12348	0.4346	0.233	197271	783.01
Hier.	80	250	0.9	23457	0.0267	0.443	1561	701.29
Hier.	80	250	0.6	23875	0.0254	0.451	2169	702.74
Hier.	80	250	0.3	22514	0.0296	0.425	1826	712.66
Flat	80	250	0.3	12601	0.5607	0.238	208274	797.39
Flat	80	250	0.6	11388	0.3107	0.215	193206	751.73
Flat	80	250	0.9	11493	0.5957	0.217	194516	766.03
Hier.	100	250	0.9	19825	0.0266	0.374	1656	656.03
Hier.	100	250	0.6	22424	0.0266	0.424	2300	689.80
Hier.	100	250	0.3	22297	0.0612	0.421	2031	726.95
Flat	100	250	0.3	11739	0.6248	0.222	177893	758.52
Flat	100	250	0.6	10556	0.6212	0.199	184516	768.56
Flat	100	250	0.9	13383	0.3609	0.253	181057	757.43

VITA AUCTORIS

Tai Yu graduated from Xidian University in P.R.China, where he obtained a B.Sc. in Electrical Engineering in 1997. He is currently a candidate for the Master's degree in Electrical and Computer Engineering department at the University of Windsor and hopes to graduate in January 2005.