UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

TriM: TRI-MODAL DATA COMMUNICATION IN MOBILE AD-HOC NETWORK DATABASE SYSTEMS

A Dissertation

SUBMITTED TO THE GRADUATE FACULTY

In partial fulfillment of the requirements for the

Degree of

Doctor of Philosophy

By

LESLIE DAVID FIFE
Norman, Oklahoma
2003

TriM: TRI-MODAL DATA COMMUNICATION
IN MOBILE AD-HOC NETWORK DATABASE SYSTEMS

A Dissertation APPROVED FOR THE
SCHOOL OF COMPUTER SCIENCE

BY

Le Gruenwald, Committee Chair

S. Lakshmivarahan

Deborah A. Trytten

John Y. Cheung

Joseph P. Havlicek

ACKNOWLEDGEMENTS

For most, it is impossible to complete a task such as this on one's own and I was no exception. Many people and organizations provided support, encouragement and guidance.

I wish to thank the members of my final committee: Drs. Le Gruenwald, S. Lakshmivarahan, Deborah Trytten, John Cheung and Joseph Havlicek. With their guidance and critique this research has been greatly improved. I would also like to thank original members of the committee that helped me get started. They are Drs. John Canning and Rex Page. A special debt of gratitude is owed to Dr. Gruenwald who accepted me into the OU Database Group late in the degree process. She spent many hours advising and directing my work. For her exceptional assistance I will always be grateful.

The OU Database Group provided me with a critique of my dissertation defense presentation. Their comments and suggestions made a significant difference in the final presentation. Their support in sharing and discussing research ideas proved vital.

Other friends have provided support and assistance along the way. Drs. Gopal Racherla and Sridhar Radhakrishnan, Barbara Bledsoe, Rene Jenkins, Daryl Hammack, Jim Summers, Jana Williams, Marlin Butcher, Kevin Pate and many others. It would take too many pages to detail the many ways in which they have assisted me. I offer them my thanks.

Brigham Young University – Hawaii Campus and Oklahoma City University provided me employment prior to completion of my degree. I appreciate their faith and confidence.

The greatest debt of gratitude, however, is owed to my family. My children: Jakob Thor, Kaitlyn Elisa Rebekah, Alexander MacDuff and Isobel Swan Carolyn have seen less of me than they deserve. Finally, and most importantly, I thank my wife. Jamie has endured the completion of three degrees over a period of nearly 16 years. I have been a student our entire married life. She has supported me and encouraged me along the way. I could not have done this without her.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF EQUATIONS

Abstract

The Mobile Ad-hoc Network (MANET) is an emerging area of research in the network and database communities. A MANET is a group of self-organizing, autonomous clients and servers that form temporary networks. A MANET allows three methods of data communication. These are: data broadcast, data query and peer-to-peer communication. The primary research in this area has been in MANET routing. Node mobility, disconnection, battery power and limited bandwidth form the constraints for MANET data communication research.

The objective of this research is twofold. First, a MANET data communication protocol, TriM (for Tri-Modal Communication), capable of providing all three methods of data communication in a single network is designed. This is the first MANET protocol capable of providing all three methods of MANET data communication. Second, a benchmark capable of evaluating MANET data communication protocols is developed. This is the first benchmark developed for the MANET environment.

TriM was designed to accommodate disconnection and reconnection to the network through periodic synchronization. Data communication was designed to provide contention free data broadcast. Each part of the protocol was designed with minimum power consumption as a goal.

The developed benchmark has three parts. These are a standard MANET architecture, data communication workload and evaluation criteria. This benchmark allows the evaluation and comparison of MANET data communication protocols and is used to evaluate TriM.

Simulation showed TriM minimized the average power consumption of servers and clients while accommodating node disconnection. The research also demonstrates the transmission ranges needed to get acceptable performance in large regions where the number of servers is limited. Simulation was also used to compare TriM to Gruenwald's Leader Selection protocol. This comparison showed Trim operated at similar and lower average power consumption rates while providing a greater range of data communication methods. Analysis of TriM demonstrated the benchmark was capable of accurately predicting the simulation performance of TriM under a wide range of scenarios.

Chapter 1

PROBLEM STATEMENT

1.1 Objective and Motivation

Mobile computing and networking have been an increasingly active area of computer science research. Within the past five to ten years, new conferences and journals have emerged and prospered in this area. For example, the ACM journal *Mobile Computing and Communications Review* is in its 7^{th} year. During the year 2003 the *Fifth International Symposium on Human Computer Interaction with Mobile Devices and Services*, *Ninth Annual International Conference on Mobile Computing and Networking*, and the *Fourth ACM Symposium on Mobile Ad Hoc Networking & Computing* as well as other related conferences were held. The Internet Engineering Task Force [12] has also set up working groups on mobility topics.

Within this rather broad category of mobile computing, research on Mobile Ad-hoc Networks (MANET) is an emerging area. While many possible avenues for research into MANET exist, most current work is centered on routing issues. For example, the *Seventh Annual ACM International Conference on Mobile Computing and Networking* in 2001 held two of its nine technical sessions to deal specifically with MANET topics. Of the six papers presented, five dealt with routing issues, while the sixth addressed the capacity of ad-hoc networks [78]. The same conference in 2003 had several papers and tutorials dealing with ad-hoc networks. These papers dealt with routing, topology, and mobility models [1].

One area in MANET that has only received a small amount of attention is data communication. Data communication is the MANET mechanism that allows servers to provide data to clients, clients to query servers and for clients to communicate with other clients. Without data communication, nodes in a MANET are independent and incapable of communication. This is the topic addressed by this research.

The primary objective of this research is to develop and propose a new data communication protocol for use in MANETs. There are three tasks necessary to accomplish this objective. First, is to identify the issues related to data communication in MANET and the applications suited to this network architecture; second, is to develop a standard benchmark for MANET protocol evaluation that includes a standard architecture, workload and evaluation criteria; and third, is to design a data communication protocol that includes all modes of MANET data communication and is well suited to the particular needs of the MANET. This protocol must then be analyzed and simulated according to the benchmark developed.

1.2 Organization

This dissertation is divided into seven chapters. The following paragraphs provide an overview of each chapter.

In the remainder of this chapter, the MANET architecture and environment will be described. Specific issues facing MANET research are also discussed. This chapter concludes with a description of the contributions of this research.

Chapter 2 reviews the literature in two areas. First, the literature associated with traditional wireless network data communication is reviewed. Areas of similarity and difference to MANET data communication requirements are discussed. This is followed by a review of the current MANET literature, particularly as it relates to MANET data communication.

The proposed MANET data communication protocol, TriM, is described in Chapter 3. There are many potential MANET architectures proposed in the literature. This research and the protocol developed will assume the architecture described in Section 1.4.

Chapter 4 deals with an issue critical to the development of a MANET data communication protocols. No standard benchmark currently exists for the evaluation of MANET data communication protocols. There is no standard architecture or workload. There is also no standard set of evaluation criteria for the evaluation MANET data communication protocols.

First, the type of applications suited to the MANET environment is addressed. The applications considered are for military battlefield use, domestic rescue and temporary business networks where an existing network infrastructure does not exist. Using previous MANET research architecture descriptions and an analysis of the anticipated MANET applications a standard MANET architecture and workload is proposed as part of the benchmark developed. This standard architecture and workload is then used by the MANET data communication protocol proposed by this research.

The criteria for MANET data communication protocol evaluation are also addressed as part of the benchmark developed in Chapter 4. The evaluation criteria of current MANET routing and data communication research are discussed. Based on previous criteria used and an analysis of anticipated MANET applications a standard set of evaluation measures is proposed. This set of evaluation criteria completes the proposed benchmark for the evaluation of MANET data communication protocols. This MANET data communication evaluation benchmark is used in the analysis of the MANET data communication protocol proposed by this research. The use of a standard benchmark will allow more consistent and meaningful comparison of current and future MANET data communication protocols.

Once the TriM protocol is described and the benchmarked developed, TriM is analyzed. This analysis is presented in Chapter 5. The analysis has two parts. First, TriM will be discussed in terms of the MANET data communication research issues described in Chapter 1. The manner in which TriM addresses the specific needs of MANET data communication is discussed. Second, TriM will be analyzed in terms of the benchmark proposed in Chapter 4. The anticipated performance of the protocol will be discussed.

TriM will also be simulated. The architecture used by the simulation will be the MANET benchmark architecture proposed in Chapter 4. All three anticipated MANET application scenarios will be simulated. A description of the simulation, the simulation results and an analysis of the simulation is presented in Chapter 6. The simulation results will consist of the parameters required by the benchmark for MANET data communication protocol evaluation proposed in Chapter 4.

Chapter 7 will contain the conclusions drawn from this research. The MANET data communication protocols presented in the research of Wieselthier, et. al. [71] and Gruenwald, et. al. [37] will be compared to the proposed protocol. However, complete comparison is not possible, as the work of Wieselthier and Gruenwald do not accommodate all three forms of MANET data communication. In addition, the calculated performance of the protocol done as part of the analysis in Chapter 5 will be compared to the results of the simulation done in Chapter 6. The dissertation will conclude with suggestions for appropriate future work in this area. Following Chapter 7 will be a bibliography and appendices.

1.3 Mobile Ad-Hoc Network Overview

A traditional mobile network consists of a fixed network of servers and clients, with a collection of mobile clients that move throughout the geographic area of the network. Within the traditional mobile network, servers have unlimited power and communicate with both fixed and mobile clients. Fixed clients also have unlimited power and are accessed over the wired network.

Mobile clients are battery powered and are accessed over a wireless connection. Two mobile clients may only communicate through a server. Mobile clients are not capable of communicating directly.

The telephone system is an example of this type of network. Within the telephone system are both fixed and mobile clients, communicating through the fixed infrastructure of the telephone service providers. Among the issues in this type of network are client power consumption, connectivity of the network, and reachability of mobile clients from a server.

In contrast, a MANET is a temporary network composed of a collection of mobile servers and clients. All nodes (client and server) are wireless, mobile and battery powered [73]. The network topology can change frequently and may change rapidly. The nodes organize themselves automatically [7]. Current MANET data communication research is silent on the ability to add nodes to a MANET throughout deployment. All have assumed a fixed set of nodes [37][46][69][70]. In contrast, commercial ad-hoc protocols such as Bluetooth do not limit the ability to add nodes to the network at any time [2b]. Bluetooth is a wireless protocol that connects both mobile and fixed nodes over a wireless network. In Bluetooth networks not all nodes are mobile and battery powered [2b]. This is not the type of network used in this research. Instead we assume that the number of nodes is fixed at network deployment. While nodes may connect and disconnect from the network throughout a MANET deployment, once initialized, the number of nodes in the network cannot be increased. This restriction is reasonable for a temporary network as envisioned for battlefield, rescue and business applications. In each case, the maximum number of nodes can be planned for and determined in advance.

Originally called Mobile Packet Radio, Mobile Ad-hoc Network (MANET) technology has been an important military research area [28]. This technology has practical use whenever a short-term network is needed and no fixed infrastructure is available. MANET is an emerging research area. For that reason, not every network described as a MANET is in fact a mobile ad-hoc network. For instance, in [32] it is not clear that the network described is anything more than a traditional mobile network.

Potential applications include communication and data service on the military battlefield, during rescue operations, and for conferences or other business uses in non-traditional locations [58]. In each case, a permanent wired infrastructure does not initially exist and building one is too slow, too expensive or impractical. The support of these military and civilian uses often requires the presence of a database to store and transmit critical mission information such as inventories or tactical information.

MANET characteristics include a preference for reactive (on-demand) routing, unpredictable and frequent topology changes and distributed (localized) control [58]. MANET primary limitations are limited bandwidth and limited battery power [58]. Within this research, it is assumed that the routing algorithm used employs reactive routing. Other researchers are developing several potential and appropriate reactive routing algorithms in parallel [6].

Traditional wireless networks involve the server in all data communication. MANET includes the traditional capabilities of broadcast (data push) and data query (data pull). MANET also allows peer-to-peer communication (peer messaging), where clients can communicate directly to other clients without the involvement of the server, unless necessary for routing [30][43]. This allows clients to communicate directly, without the use of a server, when they are within each other's broadcast area.

1.4 Mobile Ad-Hoc Network Architecture Overview

As part of the introduction, this section provides an overview of the MANET architecture used throughout this research. The nodes in a MANET can be classified by their capabilities. A Client or *Small Mobile Host (SMH)* is a node with reduced processing, storage, communication, and power resources. A Server or *Large Mobile Host (LMH)* is a node having a larger share of resources [37]. Servers, due to their larger capacity contain the complete database and all database management functions found in the database management system (DBMS). It is assumed by this research that all servers maintain copies of the same database and have full replication. The servers bear primary responsibility for data broadcast and satisfying client queries. Clients typically have sufficient resources to cache portions of the database as well as storing some DBMS query and processing modules [37].

As both clients and servers are mobile, the speed at which the network topology changes can be rapid. A variety of techniques have been proposed to assist in the routing tasks of MANET. New protocols were necessary as the protocols for fixed infrastructures and static networks do not perform well when node mobility is included [53]. A global routing structure is also not useful in MANET due to its dynamic topology and need for distributed control [53]. Work on routing is ongoing and is coordinated through the Internet Engineering Task Force (IETF) [12].

Nodes may not remain connected to the network throughout their life. To be connected to the network, a node must be within the area of influence of at least one other node on the network and have sufficient power to function. The area of influence is the region in which a node's transmission can be heard. However, the number of nodes is fixed for each MANET deployment.

In Figure 1-1, a few nodes of a typical MANET are shown graphically. It is important to note that each node has an area of influence. This is the area over which its transmissions can be heard. A LMH will initially have a larger area of influence as it generally has a more powerful battery. Some research assumes a variable powered broadcast transmission. For example, Wieselthier sets the transmission level as part of the algorithm for building the broadcast tree [71]. In this research, we assume a fixed transmission power level. Because of this, the area a broadcast transmission reaches is determined primarily by the amount of power remaining in the node's battery. As the power level decreases, the area of influence of any node will shrink. This is due to the fact that the power available to broadcast is reduced. The size of the area of influence may vary by node.

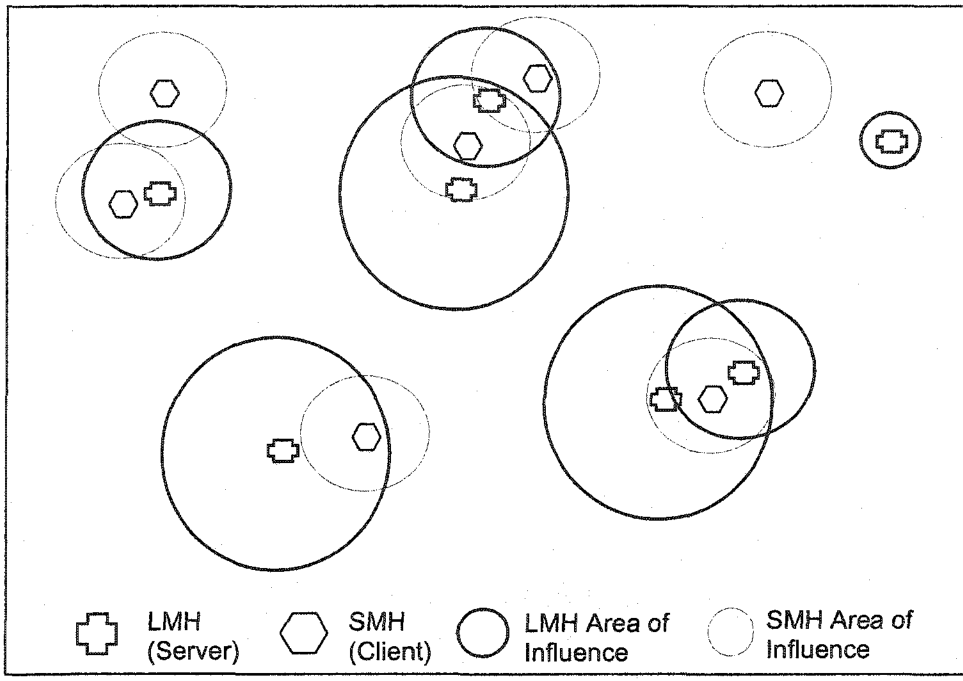


Figure 1-1 Typical MANET Architecture

Network nodes (client/server) may operate in any of three modes that are designed to facilitate the reduction in power used [45][67]:

- Transmit Mode: this is the mode using the most power. It allows both the transmission and reception of messages and consumes 3000 to 3400 mW [67].
- Receive Mode: the CPU is capable of processing information and is also capable of receiving notification of messages from other nodes and listening to broadcasts. 1500 to 1700 mW are consumed in this mode [67].
- Standby Mode: the CPU does no processing and the node has no ability to send/receive messages. The node is inactive and consumes only 150 to 170 mW [67]. Even in sleep mode, some power is consumed. The device is not completely shut down. This mode allows a node to almost turn itself off for short periods of time. The node can change to receive or transmit mode without requiring power-up or re-initialization.

A node with no remaining power, or one that is off, is not currently a part of the network.

It is clear from the description and Figure 1-1 that a node may not be reachable by another node (LMH or SMH). In other words, nodes may become disconnected from the entire network. When moving back in range of other nodes, they will become re-connected. Conversely, a node may be reachable by several LMHs or SMHs. The potentially rapid and regular reconfiguration of the network topology is routine with the MANET.

1.5 Mobile Ad-Hoc Network Data Communication Research Issues

A MANET must consider the data communication issues associated with a traditional mobile network. These issues include broadcast size and organization as well as the selection of items for broadcast. A MANET must also consider the power consumption and mobility of the server(s) as well as the power consumption and mobility of all clients. In addition, the limited bandwidth of wireless communication must be considered [58]. The issues associated with data communication in a MANET are developed in this chapter.

The data communication research issues in MANET databases center around three areas. The first area concerns the limitations of the environment (wireless, limited bandwidth, battery powered, mobile). The second area concerns the three ways in which MANET data communication may take place. Within this area, concerns due to data push (broadcasting), data pull (data query) and peer-to-peer communication (peer messaging) will be discussed separately. This section, in large part, has appeared in [35]. The third area concerns issues caused by the relative newness of MANET data communication research and the resulting lack of maturity in this area. These issues center on the lack of a standard architecture, workload and method for MANET data communication protocol evaluation.

1.5.1 Environmental Issues

The environmental limitations of the MANET environment are power consumption, mobility and disconnection, timing and data integrity. The environmental issues are discussed in the following sections.

1.5.1.1 Power Consumption

Power consumption is a concern in any mobile network. However, in traditional mobile networks, only the power needs of the clients are considered. Here, the power of the server, which provides DBMS data services, is perhaps more important as it provides data services to potentially many clients. This is the one overriding issue [55]. The primary issues related to power consumption are:

- Are power settings broadcast for servers and clients? If so, how often?
- Do server power levels affect broadcast assignments and if so, how?
- What should be done with a LMH/SMH with a low power level.
- How is power consumption distributed throughout the network?
- Does data query and peer-to-peer communication affect power consumption?
- How does node disconnection due to power depletion affect the network?

A server's power setting can be an important input into the entire process. Servers with the greatest power availability may be expected to perform the most work. However, if this information is broadcast, power is consumed.

Broadcasting is both time and energy conscious [52]. A carefully coordinated set of broadcasts can reach a large number of clients who only have to listen to get the information they need. Only if the information needed is not broadcast does a client need to query the database.

1.5.1.2 Mobility and Disconnection

An important aspect of MANET deployments is the mobility of all nodes, both clients and servers. As nodes move, a node may become disconnected from the network by being beyond the transmission range of any other node. This is independent of disconnection due to power depletion. As nodes move back into range, they become capable of participating in the network. How this mobility affects protocols designed for MANET data communication must be considered.

1.5.1.3 Timing

Regardless of the method of communication used, access time and tuning time must be considered. Tuning time is the measure of the amount of time each node spends in transmit mode. This is the time of maximum power consumption for a client. Because of that, tuning time minimization is an important goal. Servers are in transmit mode when transmitting a data broadcast and its index. Servers are also in transmit mode when responding to data queries during data pull. Servers are also in transmit mode when routing peer-to-peer requests. Clients are in transmit mode when transmitting a query during data pull and when transmitting a message or response to a message during peer-to-peer communication.

Access time measures the responsiveness of the data pull portion of a data communication protocol. Access time refers to the amount of time a client must wait to receive an answer to a database query. If access time is too long, the client may no longer be reachable from the server assigned to transmit the information. Power is wasted if a query response is missed and must be requested again.

1.5.1.4 Data Integrity

With data integrity, we are concerned with the accuracy of information stored at each node: server and client alike. This problem occurs as servers and clients move in and out of contact.

Acknowledgement messages are not appropriate in a MANET as mobility makes receipt unreliable and extra bandwidth and power are consumed.

Data replication is an important consideration. If the database is fully replicated among all mobile servers, additional power is consumed to maintain the databases. If full replication is not required or possible, other data integrity issues exist.

While data replication may not exist for an entire network, it may be possible to maintain it in disjoint partitions within the network. Partitioning of a network or database is both carefully designed and reasonably static or is considered a failure condition [47]. As servers are mobile, partitions would be necessarily dynamic. Partitioning would also not be considered a failure in a MANET, but would be normal. Data replication would add some amount of overhead to the network. This research assumes full replication of the database among all servers.

If a system does not provide full replication, the database stored at each server may not be consistent with one another. As database updates are made, not all servers would be guaranteed to receive the updates in a timely fashion.

Nodes become disconnected for a variety of reasons. This may be due to location or lack of power. The dynamic nature of MANET makes maintaining the data a challenge. Multiple versions of the same information may exist throughout the network. When portions of the network become separated for a time, keeping data accurate may become impossible.

1.5.2 Data Communication Method Issues

A MANET can communicate in any of three ways. These are data push (broadcast), data pull (data query), and peer-to-peer communication. Issues related to each communication method are discussed in the following sections.

1.5.2.1 Data Push (Broadcast)

Of all the MANET activities, data communication remains one of the high power consumption activities. When broadcasting, each node listening to a broadcast consumes much less power than is consumed by the broadcast transmitter [45].

Traditional mobile network protocols [6][17] assume that the clients can regularly submit requests to the DBMS servers. Traditional methods [17] also use frequency of request when building broadcasts. There is nothing efficient about multiple clients individually requesting the same data item. It is also not energy efficient for servers to unicast the same data individually to several clients. It is important to minimize data requests, saving power at both the server and client.

It is important to keep in mind that while broadcast is energy efficient when working with multiple nodes, it is not sufficient when a large number of data items must be delivered [38]. Data pull alone is also not sufficient [38]. Both methods, used appropriately, are necessary to achieve the greatest energy efficiency [38].

There are a great number of data push communication issues. They are discussed in the following sections.

1.5.2.1.1 Broadcast Content

The size and contents of a broadcast affect power consumption and the frequency of data queries. If the broadcast is too large, unnecessary information may be broadcast. If too little information or the wrong information is transmitted, data queries increase. In both cases, access time also increases. Traditional mobile networks solve this problem through building broadcasts based on frequency of queries [38]. Requiring multiple clients to request the same data wastes client power. However, continually transmitting a broadcast wastes server power [17]. In a traditional wireless network, the server is connected to the power grid and so continual transmission of broadcasts is not a power issue.

Mobile network research shows that an index can minimize the amount of time clients must remain active, accessing the broadcast [41]. The tradeoff is that the index must also be broadcast. The small amount of energy needed to broadcast the index may offset the large amount of energy needed by many clients to listen to the entire broadcast.

- How often the contents of a broadcast are built/changed.
- Node's data needs – as determined from data requests not served through data-on-demand or peer-to-peer communication.
- What criteria are used to determine what is included in the broadcast?
- Should an index be used as part of the data content?

1.5.2.1.2 Broadcast Allocation

If multiple servers exist in an area, who transmits what data items? The methods proposed in [37] assume a leader that coordinates the work of the server group. This is an attempt to save power by sharing the load. But must a leader be selected? A leader selection protocol consumes power as candidate leaders transmit data and then compute the leader's identity. If the number of servers in a region is small this may be an unnecessary consumption of power. Perhaps each server can coordinate based on individual knowledge of area servers and clients.

In addition to the allocation of broadcast content, the timing of broadcasts is critical. In many ways, MANET broadcasting is like the telephone party line or a bus network topology. If several servers attempt to broadcast simultaneously, there will be a collision and the broadcast of all will be garbled. This is a waste of time and power for both the servers and the clients listening to the broadcasts. If a protocol uses a lead server to make broadcast assignments, it is possible for a node's assignment to be larger than it can accommodate, based on the node's remaining power. This is not an efficient allocation. A portion of the broadcast will not be sent and that LMH will disappear from the network due to running out of power.

1.5.2.1.3 Broadcast Frequency

Too frequent broadcasts waste server power unnecessarily. Too infrequent broadcasts lead to increased client requests, wasting their power. The frequency of broadcasts may be a function of server power levels and the data request frequency of clients. Frequency of broadcasts affects both tuning time and access time.

1.5.2.1.4 Broadcast Reasonableness

This is a question of whether or not to even transmit a data broadcast. If no clients are in the broadcast area, a broadcast is meaningless and a waste of power. If only a few are in the area of influence, handling data needs interactively may be more efficient. A method to identify and track nodes in a server's area of influence is necessary.

1.5.2.2 Data Pull (Data Query)

Data pull issues center on client data needs not met by data broadcast.

- Should a request be added to the next broadcast or served immediately?
- Should a client be prohibited from querying for the same data as another client in the same area or should it simply wait for data service?
- Does the server need to know how many clients want a piece of data to determine data importance?
- How is data aged so that all requested data is eventually broadcast?
- Is it important to serve data requests even after a certain amount of time has elapsed?
- When a SMH leaves an area, do we forward the data service request – or do we rely on the SMH to determine whether it is in a new cell and know it must re-request the data?
- How do we forward service requests in the network?

If the broadcast does not satisfy the needs of a client it must obtain the data from a server (data query) or from another client (peer-to-peer). Peer-to-peer issues are discussed in the next section. While satisfying the data needs of the client, we must remain sensitive to power consumption and mobility issues

1.5.2.3 Peer-to-Peer Communication

Peer-to-Peer communication allows an additional method of data communication. In peer-to-peer communication, clients can communicate with clients directly. The issue here is the role of the different nodes in this communication method. Existing MANET data communication protocols do not address peer-to-peer communication in conjunction with the other forms of MANET data communication [37][71]. The issues associated with peer-to-peer communication are:

- Should the client be limited in the number of peer-to-peer messages over a set period of time?
- How does the server know it needs to route a request?
- Should peer-to-peer be limited to certain types of requests?
- If a request is not serviced in time, should it be added to the next broadcast?

In addition to the various MANET data communication issues, there are other items to consider. These other issues are discussed in the following section.

1.5.3 Other MANET Data Communication Research Issues

In addition to the research issues associated with the MANET environment and the issues associated with the three methods of MANET data communication there are additional issues. These occur because of the relative newness of MANET data communication research. These issues center around the architecture used to simulate MANETs in research and the methods used to evaluate MANET data communication protocols.

There is currently no standard way that MANET data communication protocols are evaluated. There is no standard architecture or workload used when testing

MANET data communication protocols. This makes it difficult, if not impossible to compare the different MANET data communication protocols.

MANET test architectures vary greatly. An example should illustrate the problem of comparing data communication protocol results. The following protocols were all developed for ad-hoc networks. Jung tested his broadcasting protocol designed for urban areas by using 256 nodes in a 200 m x 200 m grid. The existence of node mobility was not provided and is unknown [46]. Kunz, when testing his ad-hoc multicasting protocol used 50 nodes, moving 1 to 20 m/s in a 1 km x 1 km region [51]. While these protocols may have been designed for different MANET application scenarios, nothing in the literature clarifies that possibility.

MANET multicasting and broadcasting are also evaluated to different criteria. One potential measure is *broadcast effectiveness*, used by Wieselthier in his multicasting protocol [72]. This is a measure of the ratio of packets received to packets sent or the ratio of the number of nodes reached to the number of nodes a broadcast was supposed to reach. Other protocols use measurements similar to *broadcast effectiveness*, such as the *success ratio* of Xuan's broadcast protocol [76], *effectiveness* of the Kunz multicast protocol [51], *delivery ratio* used in the William's broadcast protocol [73], *packet loss* used to measure the multicast protocol of Obraczka [61], and *robustness* used by the multicast of Durst [34]. While similar, these do not measure precisely the same thing.

To make comparison even more difficult, not all broadcasting protocols measure themselves by something similar to *broadcast effectiveness*. While some studies report broadcast effectiveness, Gu and Javed use *hit ratio* [37]. The hit ratio is a measure of the number of number of data requests satisfied by a broadcast. Guo uses a measure similar to the hit ratio, called *access probability* in a broadcast protocol [38].

While a more mature research area, MANET routing algorithm testing and evaluation is not significantly better. Each research study uses their own choices to measure their protocols. For instance, the routing protocol of Das measures *fraction of packets delivered* [30], while Johansson measure routing performance by *percent of packets received* [43].

As long as MANET test architectures and workloads are non-standard and the evaluation criteria vary, it will be impossible to properly compare MANET data communication protocols. This comes down to an issue of benchmarks. To test and compare two protocols, it is necessary to test them under identical circumstances and measure their performance in the same manner. The MANET architecture and MANET data communication evaluation benchmarks are the topic of Chapter 4.

That concludes a discussion of the many issues facing research in Mobile Ad-Hoc Networks. The following sections will summarize these issues and the contributions made by this research.

1.6 Summary of Mobile Ad-Hoc Network Data Communication Issues

Mobile Ad-Hoc Networks are an emerging area of research within the mobile computing and networking area. Research in this area is dominated by routing issues and associated protocols. The issue of data communication in MANET has not been adequately addressed.

A MANET has three methods available for data communication and may use any or all of them. These methods are data broadcast, data query and peer-to-peer communication. A MANET deployment may need to use all three methods. For instance a battlefield network may need data from company and battalion servers through regular broadcasts of commonly needed data and data query for periodically needed data. Clients at the squad level may also need to communicate with other squad level clients to coordinate movement and actions. However, no existing MANET data communication protocol exists that accommodates all three methods of data communication in a MANET.

The research issues in MANET data communication research are of three general types. The first set of issues occurs because of the nature of the MANET architecture. These issues deal with mobility, connectedness, timing and data integrity. The second set of issues occurs because of the type of data communication we are doing. When performing data query, there are some issues unique to that form of communication. The same is true for data push and peer-to-peer communication. The third set of issues is temporary and occurs because data communication research in ad-hoc networks is still in its infancy. As this research area matures, the issues related to research maturity will diminish. These are the concerns associated with a standard architecture and evaluation benchmark for MANET data communication.

A successful MANET data communication protocol must allow for any and all of these communication methods while dealing with the issues associated with the MANET architecture in general as well as each data communication method.

1.7 Contributions

This research proposes a new protocol for MANET data communication. This protocol is based on earlier protocols developed in this area. However, this protocol further extends previous work by including all three methods of MANET data communication in a single protocol. This protocol allows all three methods of MANET data communication while addressing the research issues presented previously. In particular, the protocol addresses the environmental limitations of the MANET architecture while considering the issues for each data communication method.

As new protocols are developed in emerging research areas, it is also necessary to develop standards and benchmarks. This research contributes to that cause by developing and proposing a benchmark for MANET data communication that provides a standard architecture and workload and a standard set of evaluation criteria.

The analytical and simulation results of this research demonstrate the viability of a MANET that allows all three forms of MANET data communication to exist in a single MANET implementation. This should allow the development of further protocols and applications for this important type of temporary network.

Chapter 2

LITERATURE REVIEW

2.1 Introduction to Literature Review

As discussed in Chapter 1, the primary purpose of this research is to develop a MANET data communication protocol that addresses the specific needs of a mobile and battery powered network. The issues surrounding this type of network were discussed in Chapter 1.

A related issue to this project is the development of a benchmark capable of comparing MANET data communication protocols. As no such protocol exists, one is proposed as part of this research. This MANET data communication benchmark addresses architecture, workload, and evaluation needs. The literature associated with benchmarks in general and MANET architectures, workloads and evaluation criteria in particular are discussed in Chapter 4.

In this chapter, we focus the literature review on MANET data communication and related topics. The related topics to be discussed are MANET routing protocols and traditional mobile network data communication.

This literature review is divided into three sections. Section 2.2 will focus on the research related to traditional mobile network data communication. Section 2.3 will briefly discuss the research in MANET routing protocols. This is discussed, as routing is an issue that is closely related to data communication. MANET peer-to-peer communication, for instance, involves routing. Section 2.4 will then address research specific to MANET data communication. Concluding this chapter will be an overview of the literature review, which will be Section 2.5

2.2 Data Communication Research in Traditional Mobile Networks

Data communication research in traditional mobile networks utilizing databases at the server is limited to situations where only clients are mobile and battery powered. The servers are connected to the power grid and are connected to each other over a physical network. These servers often provide data service to both fixed and mobile clients. This research is primarily focused on ways to maximize mobile client battery life by improving either the organization of the data broadcast or the selection of the broadcast contents or by addressing other issues peculiar to this type of network.

Much of the research in traditional mobile networks cannot be directly applied to a MANET due to architectural differences. Chief among these differences is the battery powered and mobile server of the MANET. Still, traditional wireless networks are concerned with some of the same issues that affect MANET data communication, even when their concern is only with the mobile client.

One group of traditional mobile networks we consider are those providing data service to large populations of wireless customers. Here, the networks must balance client needs to provide a high level of service while utilizing the limited bandwidth of the wireless communication medium [17].

In 1995, Leong and Si addressed the difficulties involved in broadcasting by database servers when considered the limited bandwidth, mobility of clients, and noise on the transmission media [54]. Their goal was to provide energy efficient and time efficient access by the clients to data to preserve the limited battery power of the clients. As the servers were not battery powered, the method developed was not energy efficient for the server. The servers transmitted the data broadcast over multiple adjacent channels. The data was transmitted in a time-delayed manner, so that if a record was missed on one channel, it could be accessed on another channel [54]. The idea would be similar to having adjacent train tracks, each with a series of trains to a sequence of cities. On the first track, the trains leave in order. On the second track, a delay of one train occurs, and then the trains leave in order. If the train needed just left on track j, it will be leaving on track j+1 next, followed by track j+2. Each destination is served by one of the tracks at all or several time-sequenced intervals. This scheme is energy intensive for the server, and is therefore not appropriate in a MANET environment.

Aksoy suggests a different solution to the needs of mobile clients receiving data service from one or more stationary servers. In the Aksoy solution clients request data, which is served via broadcast. The paper focuses on the scheduling of these broadcasts [17]. As the client population is mobile, it may vary over time. Aksoy, et. al. [17] present a large-scale on-demand broadcast model called RxW (Requests times Wait). At each broadcast tick, the server chooses an item to broadcast based on the number of requests and the amount of time the original request has been waiting.

The Aksoy research has benefits and drawbacks in a MANET environment. The type of data service envisioned requires a large database. Overhead for large databases is significant in both time and space [37]. In addition, the server is constantly in transmit mode as it is broadcasting regularly. This is not a power issue in a traditional mobile network as the server is not battery powered. Still, the clients are battery-powered. Frequent client data queries have an associated power cost. The other obvious MANET issue is that the Aksoy situation only provides data pull. If the clients make no requests, there are no data broadcast transmissions.

The key idea that relates to data communication in MANET is that power savings can be achieved through the careful selection of the items to broadcast during data pull and the choices made during on-demand data service during data pull when there are multiple data items to serve.

Archarya, et. al. also address the needs of mobile clients receiving data service from stationary servers. Using the Broadcast Disk method of data broadcast in which indexed broadcasts are transmitted at regular intervals in advance of anticipated customer needs. The weakness of this data push method is that if data needs of the client are not met by a data broadcast, no method to receive this data is generally provided. Acharya addressed this problem by allowing for a back channel for client requests, which then causes the modification of subsequent broadcasts [14]. If this back channel is frequently used, the channel becomes saturated and useless [14]. If the number of nodes is large and the data broadcast does not serve most of the client needs, this saturation can be a significant problem. The importance of the data selected for broadcast and the need for periodic data-pull capabilities is underscored by Acharya's research. However, data requests are not served immediately like they would be in a data-pull environment.

Guo, et. al. [38] also work on improving the responsiveness of database service. In their approach, the server maintains a list of popular and less popular items. The popular items are continuously broadcast. If a less popular item is needed, a client may request it. This interrupts the broadcast, which continues with the data broadcast after serving the request. The server never stops broadcasting, consuming power. The idea of allowing both data push and data pull is important. However, the actual method is not appropriate for a MANET as it is energy expensive for the servers.

Yajima, et. al. [77] and Grassi [36] approach the problem differently. They try to improve database service by the organization and use of the broadcast. Yajima [77] builds broadcasts where highly correlated items are found together in the broadcast, minimizing the number of times a client must access the broadcast. The idea is similar to the idea of temporal and special locality in computer architecture. If a client needs a data item, the expectation is that highly correlated items will also be needed.

Grassi [36] uses prefetching of related items into the client cache so that they will be available locally if needed. While prefetching may shorten the time a client needs to access a data item, prefetching wastes power and space through accessing and storing broadcast items that may not be needed. The benefits from a correlated broadcast require constant processing and broadcasting by the server, leaving it constantly in transmit mode. This is not energy efficient for the servers.

There are many other protocols that have been advanced for data communication in traditional mobile networks. The previous discussion presents several, but is certainly not an exhaustive list. The common thread that binds them together is they address the problems associated with limited bandwidth, as well as the power limitations and mobility of clients. Often these methods are energy expensive for the servers. As servers are not mobile in a traditional mobile network, server mobility is also not an issue. While providing an important starting point in MANET data communication protocol design, none of the methods used in traditional mobile networks is fully appropriate in the MANET environment.

2.3 Routing Research in Mobile Ad-hoc Networks

In this section we discuss routing protocol research in Mobile Ad-hoc Networks. Routing is discussed as some data communication methods in MANET involve routing. Specifically, peer-to-peer communication requires routing when the sending and receiving nodes are not within transmission range of one another. In these cases, servers are used to provide routing services.

As both clients and servers are mobile, the speed at which the network topology changes is unpredictable and can be rapid. A variety of techniques have been proposed to assist in MANET routing. New routing protocols were necessary as the protocols for fixed infrastructures and static networks do not perform well when node mobility is included [53]. For example, some fixed networks use a global routing structure and centralized control. A MANET does not do well with a global routing structure and is best suited for distributed control [53]. Routing algorithms for tradition mobile networks are also insufficient, as they do not consider server mobility.

Several ideas have been advanced for keeping routing information current in a MANET. Periodically sending a JOIN REQUEST to the entire network has been proposed as one way to dynamically maintain routing information [53]. Another approach is to estimate node location based on received signal strength [63]. This idea was developed with the cellular telephone system in mind and requires triangulation [63]. Finally, GPS has been suggested as a method to determine the position of all nodes. Once node location is known locally, various algorithms can exploit this information to make informed routing decisions [57]. However, these methods are still global and centralized.

For the past several years, routing has been the primary area of research in Mobile Ad-hoc Networks. Much of this research is coordinated or tracked by the IETF (Internet Engineering Task Force) [12] working group on Mobile Ad-hoc Networks [6]. This particular working group has been active for several years, working on routing in the MANET.

Within the IETF MANET group, many routing protocols have been developed. First were those that used GPS or other means to track the location of every node. Until the development of Mobile Ad-hoc Networks, the location of nodes was fixed in a network, or only clients were mobile. Making every node mobile and its position unknown is a difficult routing problem. Finding some way to track the nodes made the problem somewhat simpler.

Routing algorithms are also categorized as reactive or proactive. Location based protocols are by their nature proactive. Global routing tables are built using location information. A route between two nodes can then be easily calculated. Other proactive protocols do not know geographic information, but still maintain global routing tables.

Reactive protocols do not establish a route between two nodes until it is necessary for the delivery of a packet. Servers may or may not have global knowledge of the network or of node locations. There are several reactive routing protocols under development within the IETF. Within the IETF, there is no general agreement among the protocols that a reactive protocol is preferred over a proactive one. One proposed protocol even uses a hybrid reactive/proactive approach.

However, the architecture of the MANET and the type of uses envisioned argue for a distributed control and reactive routing [58]. Global knowledge of node location is not an expected characteristic of MANETs and so routing protocols should not be dependent on this information. As a reactive routing algorithm best serves a MANET, proactive algorithms will not be discussed further. We will present several of the current protocols under development within the IETF.

Dynamic Source Routing (DSR)

The first of two reactive protocols, DSR works entirely on-demand. DSR is self-organizing and self configuring. DSR has two main modules. They are *Route Discovery* and *Route Maintenance*. Route Discovery is ongoing. The route for every packet is contained in its header. The Route Discovery module looks at all packets passing through. From this it builds a local route to other nodes. Route Discovery can also actively find a route when a known one does not exist. Route Maintenance detects and invalidates broken routes. Working together DSR can handle up to 200 nodes with high mobility [44]. The scenarios envisioned in Chapter 3 have significantly more than 200 nodes. This presents a scaling problem.

It is important to note that while it is a simple algorithm with low overhead, DSR is a very good algorithm for small networks. In several simulations against other MANET protocols, DSR has proven to be very reliable and consistently one of the top performers in a variety of scenarios [22][30][43]. In each case, the 200 node limit was not exceeded.

Ad Hoc On Demand Vector Routing (AODV)

Another reactive protocol, AODV has low memory and processing needs. It discovers routes when necessary and only maintains routes for nodes that are currently active. Routing control is localized (distributed) resulting in low network utilization, as a global routing scheme does not require updating. One feature of this protocol is that it ensures freedom from loops in the paths it maintains [62].

Zone Routing Protocol (ZRP)

A hybrid protocol, ZRP has both reactive and proactive features. For each MANET, a zone radius is selected. The size of the zone is a factor of the volume of traffic and the how dynamic the network is. Proactively, each node maintains a routing table for the zone in which it currently resides. This is maintained through periodic network messages. The protocol is reactive for messages that travel between zones. A tunable protocol, ZRP has many of the best features of both reactive and proactive protocols [39].

Some researchers have expressed the opinion that only reactive routing is appropriate for MANET [19][58]. Others merely point out the overhead of proactive routing while extolling the reduced costs of on-demand routing algorithms, such as DSR [60].

Reactive routing with distributed control has the advantage that there is no single point of failure [58]. In addition, reactive algorithms are more energy efficient [75], making them very attractive to MANET applications. However, there is nothing specific to MANET that precludes the use of proactive routing. The reactive versus proactive issue is not a subject of this research and is left to those working on routing algorithms for MANETs.

These protocols have been discussed to give an idea of variety of protocols under development. It is not an exhaustive list of the protocols recommended for MANET. However, it includes important reactive protocols under development.

Most of the current work with MANET routing protocols is to extend and improve the existing ones rather than the creation of new routing protocols. For example, Wu

[74] has shown ways to improve the IETF protocols by adding route maintenance. This will allow the protocols to invalidate routes that are not broken, but have become more costly due to node movement. Route maintenance is most successful when routing traffic is low [74].

As discussed previously, routing is important but not central to this research. It is sufficient to see that a wide variety of protocols are available. Appropriate protocols will be utilized for peer-to-peer communications.

2.4 Data Communication Research in Mobile Ad-hoc Networks

In previous sections, the literature of the two areas most closely tied to MANET data communication has been discussed. In this section we turn to the literature of MANET data communication to investigate what has been done, what problems exist, and what still needs doing.

MANET data communication may occur in three different ways: data push, data pull and peer-to-peer communication. We can use data push to transmit a broadcast to all clients in the area reached by a data server. In data pull, clients may request specific data from a data server. Finally, clients may communicate directly with other clients in a peer-to-peer fashion. The challenge in MANET data communication is in providing all three methods in an orderly manner that is energy efficient for both clients and servers. These methods should be as time sensitive as possible as the mobile nature of a MANET turns late delivery of data potentially into the non-delivery of data as the client needing the data may no longer be reachable after some period of time due to disconnection caused by either distance from other nodes or a failure due to insufficient battery power.

While research has been done involving each of these data communication methods, and pairs of these methods, no research has been done which includes all three forms of communication. To allow a MANET to have maximum flexibility in the way it communicates, all three methods should be available to any MANET deployment. Below the recent work in Mobile Ad-hoc data communication is addressed.

Some work in MANET data communication has been scenario specific. In the work of Jung, et. al. [46], broadcast in urban areas is addressed. Specifically, Jung deals with the issue of Location Dependent Queries (LDQ). These are queries where the correct response is dependent on the geographical location of the client. For example, if a client wants to know where the nearest restaurant is, the location of the client becomes an important part of the question.

A LDQ requires information on the location of the client to be known. This research proposes a general-purpose protocol where location information is not assumed. For this reason, LDQs will not be addressed. The nature and makeup of LDQs becomes deployment specific and has additional hardware requirements.

Tang, et. al. [68] adapt MANET data broadcasting to the more specialized power controlled wireless ad-hoc networks. In these networks, servers have the ability to broadcast at one of several discrete power levels. This allows the server to choose a power level appropriate for reaching the maximum number of clients while avoiding interference with other servers. This method uses a head server selection with clustering around these head server nodes [68]. These clusters are self-forming. While the multiple broadcast power levels are not included as a base assumption, the ideas of Tang are useful. The underlying idea is that a server should not broadcast more than necessary to be useful. In a MANET, broadcasting when there are no clients in your region or when other servers are already serving a client group are times when broadcasting may be non-productive for a specific server.

The work of Kunz and Cheng [51] demonstrated that tree-based routing algorithms for on-demand data service are not efficient. They suggest that a mesh-based routing algorithm, where there are multiple paths between nodes, is more efficient and resilient in a mobile environment. The idea that this paper contributes is that peer-to-peer communication should utilize routing techniques that have been specifically designed for MANETs. In the same manner, data communication techniques must also be tailor-made for MANET service.

The work of Tseng, et. al. [69] deals with a specific problem with wireless broadcast. This is the broadcast storm. They demonstrate through simulation that overlapping broadcast regions can create a significant problem in MANETs [69]. This serious problem must be considered in the design of any MANET data communication technique that involves data broadcast. As with the work of Tang, mentioned above, the underlying idea is that having servers compete for the limited wireless bandwidth wastes power and prevents clients from being adequately served.

Wieselthier, et. al. have been working together on MANET broadcast issues. Their approach is the construction of a minimum-energy tree rooted at the broadcast source [71][72]. Two algorithms called Broadcast Incremental Power (BIP) and Multicast Incremental Power (MIP) have been advanced for building these trees. The BIP builds the minimum energy tree for a broadcast, while the MIP uses the BIP algorithm, but only includes those branches necessary to reach the clients needing to receive a specific broadcast [71]. The algorithms were tested and showed that by utilizing broadcast in a mobile environment, energy savings can be achieved. Further studies with larger networks were recommended [71]. However, node mobility was not addressed.

The cost of building the tree is considered negligible by the authors as they assume the time period the tree is used will be long when compared to the cost of building the tree [71]. This would be the case for stationary nodes. However, stationary nodes would be the exception in MANET. Wieselthier accommodates "movement" with the observation that increasing transmitter power will allow them to reach nodes in new locations [72]. No potential interference between broadcasts and no need to rebuild the tree once created are considered.

The restrictions and assumptions are limiting. In addition, tree-based protocols do poorly with node mobility [38]. The problems of limited bandwidth, the need for tree maintenance, and node mobility remain.

Two algorithms to handle data push and data pull within the MANET were proposed in [37]. The first is the adaptive broadcast scheduling algorithm. Within this algorithm there are two potential ways to construct a broadcast. New items may be either added to the algorithm or may replace less important data items [37].

A global network where all servers in a region know the location and power of all other servers in the region and full replication of the database is assumed. Periodically, each server broadcasts its location and power level. This begins the broadcast cycle [37]. This is a soft real-time system. There are deadlines for data delivery. The deadlines were used to determine which data request to service although no penalty for missing a deadline was mentioned.

There is also a leader protocol that selects the server in a region with the greatest remaining power. The leader coordinates the broadcast responsibilities of other servers in its area of influence [37]. The lead server determines which portion of a broadcast each server transmits. The power level of each server drives this broadcast assignment. The server with the least power transmits the most important data items [37]. No server transmits the entire broadcast unless it is the only server in a region. After the conclusion of broadcasting, clients are permitted to query the servers. After the time period for queries, the broadcast cycle repeats [37].

This initial algorithm has a potentially large communication overhead, servers with no clients still broadcast, and less popular items may starve or be broadcast too late [37].

The second algorithm utilizes a popularity factor (PF), as suggested by Datta, et. al. [31]. The PF is a measure of the importance of a data item. The PF increases each time a request is made for a data item [37]. The amount of time since the request was made also affects the PF. If it has been too long, the need to broadcast the item may be gone. This factor is called the Resident Latency (RL) and is system and scenario specific [37]. The PF decreases whenever a request exceeds the RL value [37]. The PF is used to assist in the building of relevant broadcasts and includes RL in order to make allowances for the movement of nodes. When the PF of broadcast items is high, the probability of a broadcast that serves maximum needs increases.

If a server has not received any requests for a certain number of broadcasts, it will sleep rather than broadcast to an empty audience [37]. Finally, to localize data delivery, the lead server assigns each server the amount of data to broadcast but not the items to broadcast [37]. To deal with insufficient power levels, the servers rebroadcast the previous index and broadcast if they have insufficient power to build a new broadcast [37]. It is not clear why broadcasting old information is preferable to no broadcast at all.

This approach is still not sufficient as servers can be assigned a broadcast larger than their power levels would permit. Power and bandwidth is also wasted with duplication.

2.5 Summary of Literature Review

We have reviewed the literature associated with data communication in traditional wireless networks. This material provided important insight into the issues associated with wireless communication. The literature associated with traditional wireless networks specifically dealt with some of the bandwidth and client mobility and power issues. However, the solutions proposed were not energy-efficient for a battery powered and mobile server as exists in the MANET.

Routing protocols for MANET were then considered. This was primarily to introduce the ideas associated with MANET routing. It is clear from the discussion that new routing protocols were needed, as the routing protocols of traditional wireless networks were insufficient. In a similar manner, new MANET data communication protocols are also necessary.

Finally, recent work in MANET data communication was reviewed. Much of the current research in MANET data communication is for specific types of networks, servers, or scenarios. These ideas are helpful in the design of a general-purpose data communication protocol. However, none were sufficient on their own. In the case of the more general data communication protocols, such as those of Wieselthier and Gruenwald, specific deficiencies were noted. However, the primary deficiency is that neither includes peer-to-peer communication as part of their data communication protocol.

This review of the current literature points to the specific needs of a new MANET data communication protocol. This protocol is discussed in Chapter 3. Following the discussion of the TriM protocol, the benchmark developed for evaluation of MANET data communication protocols will be presented in Chapter 4. Currently, no standard way exists to evaluate MANET data communication protocols. This benchmark will address that issue.

Chapter 3

TriM DATA COMMUNICATION PROTOCOL

3.1 Protocol Overview

In this chapter we present the TriM protocol for MANET data communication. In this first section, an overview of TriM is given. Following this section, each stage of TriM is presented in turn. In a MANET we need to provide three modes of data communication. These are data push (data broadcasting), data query and peer-to-peer communication. Some data communication modes in TriM run sequentially while other modes run in parallel.

In Figure 3-1 we see an overview of TriM. This figure shows a single iteration. TriM will cycle through these stages repeatedly. A single time through the protocol is referred to as a service cycle (SC). Here we clearly see the relationship of each data communication mode within the various TriM protocol stages.

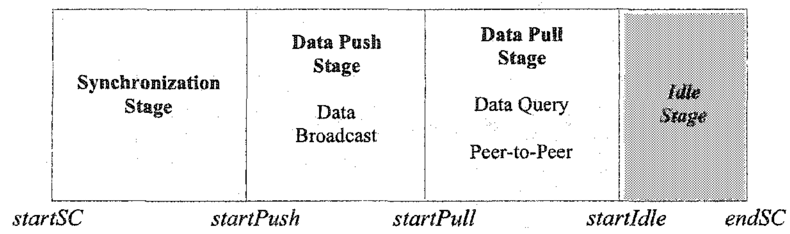


Figure 3-1 TriM Data Communication Protocol

As shown in Figure 3-1, the service cycle progresses through four stages. The service cycle begins at the start of the synchronization stage. The service cycle then progresses through the data push stage, the data pull stage and ends at the conclusion of the idle stage. The service cycle repeats continually during the life of a MANET.

Prior to the first iteration of the service cycle, the network is initialized and deployed. At this time, all protocol parameters are set. Currently, these parameters are static. Once set, they remain unchanged during the life of the network. This includes the number of LMHs and SMHs in the network.

The length of each stage of the service cycle, and other parameters are fully adjustable between different MANET deployments set up for different uses. By initializing different deployments with different parameter values, we can tune a network to different tasks. Guidelines for setting these parameters are discussed in this chapter. The primary stage variables are listed in Table 3-1.

Parameter	Description
startSC	Start of the Service Cycle & Synchronization Stage
startPush	End of *Synchronization Stage / Start of Data Push Stage*
startPull	End of *Data Push Stage / Start of Data Pull Stage*
startIdle	End of *Data Pull Stage / Start of Idle Stage*
endSC	End of *Idle Stage & Service Cycle*

Table 3-1 Primary Stage Variables

In two of the four stages, data communication can take place. These are the data push stage and the data pull stage. The synchronization is necessary to allow LMHs/SMHs to synchronize and detect the other nodes in their immediate vicinity. The idle stage allows the setting of a period of time during which all nodes are inactive. This gives the network designer the ability to set the frequency of data communication within the network. By setting this parameter carefully, we can avoid too frequent repetition of broadcasts or the other energy expensive portions of data communication. The service cycle repeats until the network is taken out of service or all nodes fail. Nodes may fail due to battery depletion or other causes, such as hardware failure.

Each node (LMH and SMH) has three power settings [56][67]. These three power settings are transmit, receive, and standby. A node is in receive mode during normal operation. In this mode it can listen to broadcasts and other transmissions, retrieve packets from transmissions, and perform all normal processing including data storage. When a node needs to transmit information, it must switch to transmit mode. A node will stay in transmit mode for as short a period of time as possible, as this mode uses the maximum amount of battery power. When a node has no data communication responsibilities, the protocol switches the node to the standby mode. This mode uses the minimum amount of power.

Throughout the TriM protocol description there are times when LMHs and SMHs are said to be in standby. This necessarily assumes that the node in question has no other data communication tasks to perform. TriM does not try to determine if there are other non-communication tasks to perform. The standby designation, therefore, only implies that there are no data communication requirements needing a more active level of participation from the client or server. When other activities are required of clients and servers, they may be performing these non-communication tasks during periods where the communication protocol indicates that they are in standby mode. Determining the other tasks a node might need to perform is not a responsibility of this protocol.

3.1.1 Network Initialization

There are four stages in the TriM protocol. Within each stage are one or more tasks. These parts are described separately and then are put together into a coherent whole. For each stage, the issues related to LMHs and SMHs are discussed separately. Three of the stages are active stages. These are synchronization, data push and data pull. The idle period is an inactive stage.

Network initialization is accomplished when deploying a MANET. The network designer determines the length of each of the protocol parameters according to the needs of the network and the expected characteristics of that particular deployment. Guidelines for setting these parameters are discussed in the next section. Typical applications for a MANET are battlefield deployment, business meetings in non-traditional locations, and rescue operations. A MANET is expected to be a short-term network. Each of these applications mentioned is short-term, running for hours or days. A MANET is not a long-term network solution. If a long-term solution is necessary, the network servers would be connected to a stable power source and portions of the network might be connected to an existing wired network.

Network initialization involves a variety of parameters. The parameters are detailed in Tables 3-1 to 3-4, occurring throughout this chapter. Typical values for some of these parameters are shown in Chapter 4 for a variety of MANET applications. Other parameters are mentioned as part of the protocol. In addition to these tables, all variables associated with this protocol are also listed together in Appendix A. Guidelines for initializing network parameters are discussed in Section 3.1.2. However, these are guidelines only. A network developer can choose other values for these parameters if they determine that special circumstances exist that make different values more appropriate. In this way, the protocol allows the designer maximum flexibility while providing guidance. Each type of node in the network is initialized using the same parameter values. All LMHs have the same parameter values and each SMH has the same parameter values. LMHs and SMHs may have different parameter values. These values remain the same throughout the MANET deployment, allowing synchronization during the lifetime of the network.

It is possible that the parameter values may be chosen poorly, leading to an inefficient network. Inefficiency refers to the MANET data communication benchmark evaluation criteria discussed in Chapter 4. If performance is an issue, the network can be reinitialized with updated parameters.

Parameter	Description
startSC	Start of the Service Cycle
synchLMH	Length: LMH Synchronization
synchSMH	Length: SMH Synchronization
synchLen	Length: Synchronization Stage: syncLMH + syncSMH
bcastPrep	Length: Broadcast Preparation
bcastLen	Length: Broadcast Period
pushLen	Length: Data Push Stage: bcastPrep + bcastLen
pullLen	Length: Data Pull Stage
idleLen	Length: Idle Stage

Table 3-2 Primary Network Protocol Parameters

The value of each of these variables is known by each LMH and SMH allowing them to monitor where in the service cycle they are at any time and to synchronize with other LMHs and SMHs. During a single MANET deployment, these values remain unchanged. Each node (LMH and SMH) individually keeps track of time, calculating the stage of the service cycle based on the value of each parameter, stored locally at each LMH and SMH.

The parameters of Table 3-2 are related to the stage variables of Table 3-1. For example, *startPush* = *startSC* + *pushLen*. The same relationship exists between all of the stage variables and the stage lengths.

3.1.2 Network Parameter Guidelines

The designer of each MANET deployment is responsible for setting the parameters given in Table 3-1 and Table 3-2. Section 3.1.2 and associated subsections provide guidelines for setting these network parameters. The parameters associated with each stage are discussed in turn. First we will discuss the parameters associated with the synchronization stage followed by the data push stage, data pull stage, and idle stage.

3.1.2.1 Synchronization Parameters

The first stage in the service cycle is the synchronization stage. The parameters associated with this stage are shown in Table 3-3.

Parameter	Description
synchLen	Length: Synchronization Stage: syncLMH + syncSMH
synchLMH	Length: LMH Synchronization
synchSMH	Length: SMH Synchronization
$transmit_{LMH}$	Time for one LMH to transmit its unique ID and location
numLMH	Number of LMHs
$transmit_{SMH}$	Time for one SMH to transmit its unique ID and location
$density_{SMH}$	Average number of SMHs reachable by each LMHs transmission.

Table 3-3 Synchronization Stage Parameters

The first parameters are *synchLen, synchLMH,* and *synchSMH*. The simplest is *synchLen*. These parameters are also listed in Table 3-2 as they control the length of this protocol stage. The length of the synchronization stage is the sum of its two parts. These are the length of the LMH synchronization and the length of the SMH synchronization.

$$synchLen = synchLMH + synchSMH \qquad (3\text{-}1)$$

Two things determine the value of *synchLMH*. The first is the amount of time needed for each individual LMH to transmit (or broadcast) its unique ID and location, called $transmit_{LMH}$. Throughout this document we use the word transmit to refer to a node transmitting or broadcasting information. We reserve the word "broadcast" to refer to the set of data items that are transmitted during the data push stage of the protocol. The other value is the number of LMHs, called *numLMH*. To allow each LMH to transmit the necessary synchronization data, *synchLMH* must be long enough. Otherwise, not every LMH can transmit the necessary information during the synchronization stage.

$$synchLMH = transmit_{LMH} \times numLMH \qquad (3\text{-}2)$$

Each LMH has a unique ID. These IDs are numbered from 1 to *numLMH.* For each LMH, the number of LMHs that have smaller IDs is ID – 1. Each LMH will transmit its synchronization data in ID order, preventing transmission collisions. Each node waits enough time in the synchronization stage for all LMHs with smaller IDs to transmit their information. The time to wait can be calculated by each LMH as (ID –

1) x $transmit_{LMH}$. The total time for all LMHs to transmit their synchronization data is *synchLMH*.

Setting the value for *synchSMH* is somewhat more challenging. It is necessary for LMHs and SMHs to know that there are SMHs near enough to hear a transmission. The other nodes (clients and servers) also need this information during data query and peer-to-peer communication. However, the protocol does not require that a LMH know all of the SMHs in its transmission reach. Therefore, it is not necessary to set *synchSMH* long enough for each SMH to transmit in turn.

To set *synchSMH* the network designer must determine the average number of SMHs that will be reachable by each LMH's transmission. Enough time needs to be allowed for SMHs to synchronize. However, it is too time-consuming to reserve individual time for each SMH in the network, as was done for the LMH. A business network may have a thousand SMHs. This average number of SMHs reachable by each LMH is referred to as the SMH density and is represented by the variable $density_{SMH}$. This estimate is based on the number of SMHs, LMHs and the network scenario. For instance, in a battlefield scenario where a LMH typically services a squad of 6 SMHs, $density_{SMH}$ might be set to 6.

The other variable needed to set *synchSMH* is $transmit_{SMH}$. This variable represents the amount of time needed by a SMH to transmit its unique ID and location. The unique ID and location are used during peer-to-peer messaging. Using these two numbers, we get an estimate for *synchSMH*.

$$synchSMH = density_{SMH} \times transmit_{SMH} \tag{3-3}$$

If we take the amount of time for each SMH to transmit and multiply it by the number of SMHs expected, we get an estimate for *synchSMH*. It is impractical to reserve independent time for each SMH to transmit synchronization data in turn as there are potentially hundreds of SMHs. However, for peer-to-peer message routing to work reliably, the location of each SMH is needed. When insufficient time for all SMHs to transmit is allocated, old location values from previous synchronization stages is used. As this information becomes less outdated over time due to node mobility, peer-to-peer message delivery can become less reliable over time. For data push it is sufficient for LMHs to know of the presence of SMHs. The precise location of SMHs is not important to the data push stage.

3.1.2.2 Data Push Parameters

The data push stage consists of creating and transmitting a broadcast index and the associated broadcast. The parameters used are shown in Table 3-4.

The first three parameters were previously listed in Table 3-2. These are *bcastPrep*, *bcastLen*, and *pushLen*. The first two, *bcastPrep* and *bcastLen*, control the length of each portion of the data push stage. The parameter *pushLen* is nothing more than *bcastPrep* + *bcastLen*. The other five parameters are used to set the value of *bcastLen*.

The item parameters are the number of items in the static portion of the broadcast ($items_{stat}$) and the maximum number of items allowed in the dynamic portion of the broadcast ($items_{dyn}$). It is possible that not every item desired will fit into the dynamic portion of the broadcast. Setting the maximum size of the dynamic portion of the broadcast prevents the broadcast from becoming overly long.

As the length of a broadcast increases the likelihood that SMHs have traveled outside the transmission range of a LMH increases. If the broadcast becomes overly long, power is consumed to transmit data that may no longer reach or have value to the SMHs. A SMH can request data again in the later Data Pull stage if it remains important. The dynamic portion of the broadcast exists to serve a small number of missed data items.

Parameter	Description
bcastPrep	Length: Broadcast Preparation
bcastLen	Length: Broadcast Period
pushLen	Length: Data Push Stage: bcastPrep + bcastLen
$items_{stat}$	Number of items in the static portion of the broadcast.
$items_{dyn}$	Maximum number of items in the dynamic portion of the broadcast.
$transmit_{idx}$	Time to transmit each entry in the broadcast index.
$transmit_{data}$	Time to transmit each data item in the broadcast.
numLMH	Total number of LMHs in the network.

Table 3-4 Data Push Stage Parameters

$transmit_{idx}$ and $transmit_{data}$ are two additional parameters used. $transmit_{idx}$ is the length of time it takes to transmit each entry in the broadcast index. Each entry in the index is assumed to be the same length, requiring the same amount of transmission time. $transmit_{data}$ is the amount of time needed to transmit each data item in the broadcast. Each data item is also assumed to be of the same size and the same transmission time for the purposes of this research. Transmit, or transmission time, is calculated using the bandwidth of the transmitting node and the size of the item being transmitted.

The two primary stage parameters are *bcastPrep* and *bcastLen*. *bcastPrep* is set as the amount of time needed to build the index for a broadcast of maximum size. The maximum broadcast size is $items_{stat}$ + $items_{dyn}$. The other stage parameter, *bcastLen*, needs to be large enough to permit the transmission of a broadcast of maximum size by each of the LMHs in turn. We calculate the maximum broadcast length by calculating the time needed for each LMH to transmit an index of maximum size and a broadcast of maximum size. This is shown in Equation 3-4. We allow time for each LMH to transmit, as the expected number of LMHs is small. In the literature review of Chapter 2 and the benchmark discussed in Chapter 4, the maximum number of LMHs anticipated is 20 in the military scenario. The number of LMHs is much smaller in the business and domestic rescue scenarios. The importance of data broadcast suggests that the servers transmit sequentially, rather than in parallel. This prevents collisions in the limited bandwidth available. This also allows SMHs to potentially hear several broadcasts. If a SMH is within transmission range of more than one LMH, it will be able to receive each transmission as they occur at different times, without collision. As each broadcast can have different dynamic items, this can be beneficial to the SMHs. A SMH will listen to each LMH index transmission and then will listen to the broadcast or sleep until the turn of the next LMH. During the transmission of a LMH, other LMHs wait in sleep mode.

$$bcastLen = [\,(items_{stat} + items_{dyn}) \times (transmit_{idx} + transmit_{data})] \times numLMH \qquad (3\text{-}4)$$

3.1.2.3 Data Pull Parameters

During Data Pull, we need to allow for SMHs to transmit and receive several messages or queries. The number to allow on average is a decision of the network designer and will vary by situation. However, some guidelines will be given in this section. The data pull parameters are shown in Table 3-5.

Parameter	Description
pullLen	Length: Data Pull Stage
reqFreq	Average number of requests made by a SMH.
serve	Maximum time to serve one data request.
$density_{SMH}$	Average number of SMHs reachable by each LMH's transmission

Table 3-5 Data Pull Stage Parameters

Data pull has the fewest parameters of the active stages. However, the data pull stage may be very busy as there are many things potentially taking place.

Other than the stage parameter *pullLen*, there are only two other data pull parameters. The first parameter is *reqFreq*, which is the average number of requests that the network designer wants each SMH to be capable of making during a single service cycle. The other parameter, *serve* is the average amount of time necessary for a LMH to serve one data request. Together, these two parameters allow us to make a good estimate for *pullLen*. We add a factor, to allow multiple SMHs time to use the limited network bandwidth, by again using the parameter $density_{SMH}$.

$$pullLen = (reqFreq \times serve) \times density_{SMH} \qquad (3\text{-}5)$$

3.1.2.4 Idle Parameters

The idle parameter, *idleLen*, is set to allow the network designer to put a delay between service cycles. This allows the designer to determine the frequency of the service cycle. During the idle stage, all nodes are in sleep mode.

Parameter	Description
idleLen	Length: Idle Stage

Table 3-6 Idle Stage Parameters

We have now discussed the network initialization and the parameters that must be set during this stage of MANET deployment. Network initialization occurs once, and then the network enters a continuous iteration through the service cycle. We now discuss each stage of the service cycle in turn. The service cycle stages are discussed in the order that they occur.

3.2 TriM Synchronization Stage

The purposes of the synchronization phase are:

- Allow each LMH and SMH to become aware of other nodes (LMH and SMH) in their transmission area.
- Determine the location of other nodes (LMH and SMH) in transmission area using location information transmitted during synchronization.
- Synchronize all nodes (LMH and SMH) to the Service Cycle.

The synchronization stage has two parts. The first part is restricted to the transmission of information by the LMHs. LMHs transmit their unique ID and their current location. There are generally fewer LMHs and their individual presence is critical to the protocol. Sufficient time will be allocated during LMH synchronization to allow all LMHs to transmit their information. The location of LMHs is used by SMHs during data query to select the nearest LMH to query.

The second stage is for transmission of information by SMHs. Each SMH transmits its unique ID and location. For data push, it is sufficient to know that SMHs are present. During peer-to-peer messaging, location information is used to route messages. If all SMHs do not get to transmit, location information from previous service cycles is used for routing. Over time, this data may become unreliable.

However, the protocol attempts to allow sufficient time for all or most of the average number of SMHs in a transmission region to be able to transmit their identifying information. The performance of data pull and peer-to-peer communication is improved by complete information about SMHs in each node's broadcast region.

The synchronization stage is important. By regularly synchronizing all nodes, each node will be in the same stage of the protocol at the same time. This allows the synchronization of data broadcasts, preventing contention over the limited network bandwidth. The location information is also used during data query and peer-to-peer messaging. Recent information on location is necessary for the data pull stage to work efficiently.

The results of the synchronization stage are considered valid for an entire service cycle. Continuing to resynchronize throughout the Service Cycle would require more transmitting and processing at a cost in power dissipation. If a SMH misses the relevant portion of the broadcast it still has the ability to acquire the needed information during data pull or in the next service cycle. A SMH can get the data from any LMH it can communicate with as all LMHs have a fully replicated database.

It is asserted that considering node synchronization to be valid for the entire service cycle is an acceptable condition. First, the stage that relies the most on synchronization is the data push stage. This stage immediately follows synchronization. Synchronization is important as it prevents LMHs from transmitting at the same time, wasting power and the limited network bandwidth. Second, when one considers the type of applications served by a MANET, it becomes clear that mobility is only part of the story. In a business meeting, the LMHs can be distributed throughout the space. While SMHs are mobile, they move slowly. Consider a trade show. People move throughout the day, but not rapidly. While topology changes throughout the day, it is not rapid. In the case of a rescue situation, nodes also move slowly and in a clearly defined region. The application of greatest speed of mobility is the battlefield situation. However, this situation is the least random in some respects as it is anticipated that nodes will be moving at similar speeds and in similar directions. Their position relative to each other will be stable for lengthy periods of time. This will provide a stable topology over short periods of time.

During the life of the network, it is possible for any node (LMH or SMH) to become disconnected from the network. This will be determined during the synchronization stage. If a node detects no other LMHs or SMHs during synchronization, the node will switch to standby during the remainder of that service cycle. Shortly before the next synchronization stage, this node will become switch to receive mode and try to resynchronize. The synchronization stage is shown graphically in Figure 4-2.

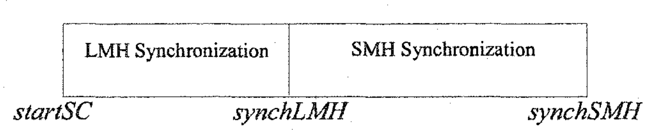


Figure 3-2 Service Cycle Synchronization Stage

In the following section the tasks for LMHs during the synchronization stage are discussed. This includes the LMH tasks during both portions of synchronization. This is followed by a section that details the tasks for SMHs during the synchronization stage, both during LMH synchronization and SMH synchronization. When not transmitting, all nodes (LMH and SMH) remain in receive mode to allow them to receive and record any synchronization data transmitted within their hearing.

3.2.1 LMHs - Synchronization Stage

During the first part of the synchronization stage, each LMH transmits its unique ID and current location. Each LMH knows the number of LMHs that were deployed during network initialization. The unique IDs are numbered from 1 to *numLMH*. Each LMH transmits its information in turn, waiting the appropriate period of time before transmitting its information. The importance of the LMH information to the protocol prohibits transmission in parallel. Collisions in the limited bandwidth of wireless networks could cause the loss of information and the failure of neighboring LMHs from knowing about one another.

The required amount of time for a LMH to wait is determined by the number of LMHs having IDs smaller than its ID and the time to transmit its ID and location ($transmit_{LMH}$). The amount of time a LMH whose identifier is ID must pause is:

(ID – 1) x $transmit_{LMH}$.

Every server LMH and client SMH will run this algorithm

All SMHs remain in Receive Mode throughout, recording the ID and location of any LMH heard.
All LMHs not transmitting remain in Receive Mode, recording IDs and Locations heard.

At time = startSC
For (i = 0 to Num_{LMH})
LMH_i Switches to Transmit Mode and transmits ID and Location

Figure 4-3 Algorithm for LMH Period of Synchronization Stage

The total time for all LMHs to transmit is *synchLMH* which *can be calculated as numLMH* x $transmit_{LMH}$. *numLMH* represents the total number of LMHs in the network, while ID-1 is different for each LMH. For example, if there are four LMHs in a network, they are numbered 1, 2, 3, and 4. *numLMH* is 4 for this network. For LMH 1, there are 0 previous LMHs (1 – 1). In the same manner, LMH 4 has three previous nodes (4 – 1). They are nodes 1, 2, and 3. The behavior of the nodes during the LMH portion of synchronization is shown in Figure 3-3.

3.2.2 SMHs - Synchronization Stage

After completion of the LMH portion of the synchronization stage, the SMH synchronization period begins. The SMH portion of which occurs at *time* = *startSC* + *synchLMH*. Each SMH i waits for a clear channel. If the channel is clear, the SMH transmits its unique ID and location. The ID is used to identify nodes and is used, along with location, during peer-to-peer message routing.

If two or more still attempt to transmit at the same time, a collision occurs. If a collision is detected, both stop transmitting and try again after a random pause. All SMHs transmit is their unique ID and location. Those hearing a transmission (LMH and SMH) store the ID and location of each SMH heard. The behavior of LMHs and SMHs during the SMH synchronization period of the synchronization stage is shown in Figure 3-4.

Each LMH and SMH in the network has only information about the nodes within transmission range. The decisions made by the data communication protocol are local, and are made with this incomplete information. It is assumed that a routing protocol exists for peer-to-peer message routing, using the information located at the set of all LMHs. If such a routing protocol does not exist, a SMH would be limited to one-hop message routing. This means that messages could only be sent to other SMHs it had detected or SMHs detected by LMHs within range of the transmitting

SMH. No routing more than one-hop could be supported.

Every client SMH and server LMH will run this algorithm

At time = startSC + synchLMH
While (time ≠ startSC + synchLMH + synchSMH)
All SMHs transmits ID and location after detecting a clear channel.
All LMHs / All SMHs listen to each transmission, within range
All LMHs record SMH IDs and locations heard
All SMHs record SMH IDs and locations heard

Nodes are in Transmit Mode when transmitting and Receive Mode otherwise.

Note: If all SMHs have not transmitted after *synchSMH* time has elapsed, the stage is still over.

Figure 3-4 Algorithm for SMH Period of Synchronization Stage

3.3 TriM Data Push Stage

The second stage of the service cycle is the data push or broadcast stage. This stage precedes data pull for several reasons. The data push stage occurs first so that the maximum number of potential data needs can be served, before a LMH becomes too weak to transmit data. There is limited bandwidth in a MANET. Separating data push and data pull reduces the contention for this limited resource and prevents a SMH performing data pull from interfering with the transmission of a broadcast. In addition, the data needs of a SMH may be satisfied by the broadcast. This would eliminate the need for a SMH to perform data pull operations, saving power by eliminating some SMH and LMH transmissions.

Data broadcast has the potential to satisfy the greatest number of information needs at the smallest possible cost. If data pull were required to serve all data needs or occurred first, several SMHs might request the same data items, and this data might be transmitted at different times by several LMHs. By eliminating the need for each SMH needing a data item to transmit a query for that item, power is saved at the SMH. By not having to transmit the same dynamic data item multiple times, power is saved at the LMH.

In addition, data push may eliminate many SMH data requests by satisfying at least a portion of the data needs of the SMH. When a broadcast satisfies the data needs of a SMH, the SMH can operate at a reduced power level during the data pull stage of the service cycle.

The data broadcast will be composed of both a pre-selected set of data items and a set of dynamically selected items. The dynamically selected items will be those data requests that a LMH could not service during the previous service cycle's data pull stage. The dynamic portion of the broadcast can be different for every LMH, depending on the data requests made by SMHs in its region that went unserved. During the initial service cycle, following deployment, this dynamic set of items will be empty. In the following sections we will discuss the role of LMHs and then SMHs in the data push stage.

When considering the scenarios discussed in the previous chapters, we see that the movement of LMHs/SMHs is not erratic and nodes will remain in similar relationships to each other with respect to direction and distance for lengthy periods of time. The changes in direction and distance as well as power level of all nodes are slow and gradual rather than abrupt. Even when changes occur more rapidly, their effect on the data push stage is limited as it is the first stage following synchronization.

3.3.1 LMHs – Data Push Stage

The decision to transmit a broadcast is a local one, made by each LMH. The contents of the broadcast are also partially determined by each LMH. The fixed portion of the broadcast is the same for each LMH. The static portion remains constant for all LMHs during the entire life of the MANET. While the items included are static, their values may change from one service cycle to the next. The dynamic portion of the data broadcast will vary, depending on the unserved needs of the SMHs within transmission range of each LMH during the previous service cycle. These are data pull requests that went unserved; generally due to time limitations on the data pull stage. The autonomous and mobile nature of this self-organizing network suggests independent LMHs. This also eliminates the need for and energy consumption of a leader selection protocol.

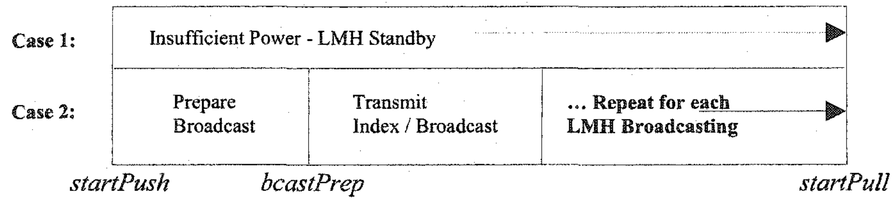


Figure 3-5 LMH Broadcast Stage

Figure 3-5 shows the broadcast portion of the service cycle for LMHs. In Figure 3-5 are shown the two possible situations facing a LMH deciding what to do. In the first of these situations, the server has insufficient power to transmit an index and data broadcast. The LMH will go into standby mode and serve what data queries it can in the data pull stage, assuming no other non-communication tasks demand the server's attention. In the other case, the LMH will broadcast in turn, in the same order that LMH synchronization occurred, based on the LMH ID. Each LMH enforces this restriction individually. Each LMH will wait for ID-1 broadcasts of maximum length before beginning its broadcast transmission. One purpose of the synchronization stage is to synchronize LMHs so that this data push synchronization is possible.

Case 1:

The first of the two situations is when a LMH has insufficient power to transmit the index and data of a broadcast. The LMH will go into standby. What power remains will be used to serve any necessary data queries in the data pull stage.

Case 2:

The second possibility is that the LMH has sufficient power to transmit a broadcast. A LMH does not necessarily know if any SMHs are reached. A SMH could be too far away from a LMH for the LMH to hear the SMH's synchronization while remaining close enough to hear a broadcast transmission. SMHs have a shorter transmission range. A LMH will always transmit a data broadcast if sufficient power remains. In this case, two activities occur. The maximum length of these activities is determined by two parameters - *bcastPrep* and *bcastLen*. During the period of length *bcastPrep* a LMH determines what to broadcast. Each broadcast is composed of an index and two data parts. The first data part consists of static data items to transmit. These are items that are transmitted in every broadcast during the life of the MANET. The values of these items may change. For example, data items that provide the status of different mission assets might be selected. The static items are determined by the network designer, and may be different for each MANET deployed but remain constant throughout a particular MANET deployment. There is also a dynamic data portion to the broadcast. By dynamic we simply mean that these data items are not automatically included in every broadcast. These are items that were not served by the LMH during the previous service cycle's data pull stage. These items will vary by LMH.

During the next portion of the data push stage, of length *bcastLen*, the index and data are transmitted by each LMH in turn. Each LMH transmits the index a single time at the beginning of its transmission followed by the static data items and then the dynamic data items. By keeping the static data portion as small as possible and limiting the size of the dynamic data portion a network designer can limit the length of the data push stage. By keeping the transmission time as short as possible, battery life is extended. Guidelines for these parameters were discussed previously in Section 3.1.2.2.

For the initial broadcast, only the static portion of the broadcast is transmitted. The dynamic portion will be empty. If a situation exists in which there are no static items to broadcast, the broadcast will be made up entirely of the dynamic portion of the broadcast.

If all known data needs are satisfied by the end of the previous service cycle's data pull stage then the dynamic data portion of the next service cycle's broadcast will contain no data items. In this case, only the static data portion of the broadcast will be transmitted.

If a particular MANET deployment requires no broadcast capabilities, *bcastPrep* and *bcastLen* can be set to 0, and this stage will be effectively eliminated. For example, a trade show may have nothing that needs to be broadcast. In this situation, the network will wait for data requests, and serve them as they arrive. Regardless of the presence or absence of static and dynamic data items, an index will always be transmitted at the beginning of a broadcast.

The static data items are predetermined for a particular MANET by the network designer. The items selected should be those that the majority of the SMHs need updated on a regular basis such as status of network assets, and mission deployment parameters. This prevents the SMHs from needing to request this data during the data pull stage, which is more costly as it requires transmission by both a SMH and a LMH. In both the static and dynamic portions of the broadcast, the value of data items may change. Static and dynamic are only meant to indicate if a data item is a permanent part of the broadcast. The index and the transmission do not differentiate between static and dynamic items. This is a designation used in the building of the broadcast.

As several LMHs may broadcast in the same region, duplication of the broadcast static portion is a waste of power. To some extent, this cannot be prevented. A SMH may be in the transmission range of several or only one of the LMHs, depending on its geographic location. However, duplication and the cost of duplication are reduced in several ways.

First, each LMH determines for itself what to include in the dynamic portion of the broadcast. A LMH will maintain a list of those data items requested in the previous service cycle that it was unable to service. This may occur because the number of items requested was especially large or because there was insufficient time remaining in the data pull stage to respond to the data request. The broadcast may vary from one LMH to another in the dynamic data items transmitted because they are serving the data needs of different SMHs. This saves power in at least two ways. First, the SMH will receive the data previously requested without the need to resubmit a data query. Second, the dynamic data items selected will be those items needed recently by SMHs in the broadcast region of the LMH. This saves power by not requiring SMHs to retransmit a data query. While the contents of a broadcast may vary among LMHs, the network parameters are constant for all LMHs/SMHs so that the network may remain synchronized.

The network designer can also affect the power consumption rate of the data communication protocol by the designer's selection of an appropriate and small set of static data items and their determination of appropriate network parameters. However, the protocol does not enforce any specific size.

If a broadcast is necessary, the index and broadcast are transmitted in LMH ID order. This restriction is enforced locally at each LMH. The synchronization stage allows LMHs to be coordinated time wise within the overall service cycle. Each LMH is allocated a broadcast slot of maximum length. This length is the amount of time needed to transmit an index of maximum size and a maximum number of static and dynamic data items. Data push is very important in servicing the needs of mobile clients, as it is energy efficient. To make certain that all LMHs have an opportunity to transmit their broadcast without interference from other LMHs on the limited network bandwidth, each is allocated an independent time slot. LMHs do not listen to the broadcast transmissions of other LMHs. Except for the time the LMH is active to transmit; the LMH may be in standby mode.

As the size of the MANET broadcast is meant to be of minimal size, a single transmission of the index is preferred as transmission of the index takes time and consumes power. The most common indexing scheme in MANET protocols is called (1,n) indexing [41]. (1,n) indexing is a technique that broadcasts the index n times during a broadcast transmission. After (1/n) of the data items are transmitted, the index is repeated [42]. As broadcasts need to be kept short due to power considerations, the index will be transmitted only a single time. Because of this, (1,n) indexing is not necessary. Frequent retransmitting of the index for a short broadcast greatly impacts the total transmission time, wasting battery power.

Every server LMH will run this algorithm

time = startPush
If (Insufficient Power) Standby
else
Build Index and Broadcast from Static and Dynamic data items.
After bcastPrep time has expired, standby until turn to transmit.

For (i = 1 to Num_{LMH})
LMH_i switch to Transmit Mode transmits index, static and dynamic data items.
LMH_j ($i \neq j$) Standby.

Figure 3-6 Algorithm for LMH Period of Data Push Stage

In [42] it was shown that (1,n) indexing provided better access time than a single index in a traditional wireless network. However, this technique does not consider the effect of multiple index transmissions on server battery power as the server in traditional mobile networks is not battery powered. The trade-off here is a potential increase in access time for a savings in LMH battery consumption.

The behavior of LMHs during the Data Push Stage is shown in Figure 3-6.

3.3.2 SMHs – Data Push Stage

The SMHs, like the LMHs have two potential situations during data push. If a

SMH detected no LMHs during the synchronization of the current service cycle, it is assumed that it will also not hear any broadcast transmissions. In this case, the SMH can sleep for the entire data push stage, conserving energy. The SMH behavior is shown in Figure 3-7. This figure demonstrates the different possible behaviors for a SMH in the broadcast stage. As mentioned previously, the designation to sleep only implies that there are no data transmission tasks to perform.

Each SMH knows from the synchronization stage which LMHs will transmit in their region. The SMHs can then tune into each broadcast index. The SMH will receive the static portion from any of the LMH transmissions it receives, but need only listen to the static portion once, as they are all identical. A SMH will also check the index for any needed dynamic data items. It will use the index to determine when the data item will be transmitted. The index contains a list of all data items that will be transmitted as a part of the broadcast, and the order in which they will be transmitted.

A SMH need only listen to transmitted indices, the static data portion once and dynamic data items of interest. To listen to these items, the SMH must be in doze mode. The remainder of the time, the SMH may be in sleep mode. In this protocol, each LMH transmits the entire static portion of the broadcast.

By comparison, in the protocol of Gruenwald [37] each LMH transmits only a portion of a broadcast as determined by a leader selected as part of their protocol. This idea was not followed for several reasons.

First, a leader selection protocol takes time and power. The selection protocol involves some amount of transmission between LMHs. During this time, nodes continue to move and power is consumed at a high rate as transmissions occur in active mode.

Second, it is easily possible for a SMH to be within the transmission range of only some of the LMHs transmitting what combined is the entire broadcast. When this occurs, a SMH does not receive the entire broadcast.

Third, the transmission of requested data is also divided among LMHs in a region, as determined by the leader. However, the LMH assigned to transmit requested data may not be the LMH nearest the requesting SMH. In this case, the SMH may not even hear the requested data item.

However, there is a cost to have every LMH transmit the entire broadcast. This is one reason the static portion of the broadcast needs to be minimal. However, the benefit is that any node that can hear a broadcast will be able to hear the entire static portion of the broadcast.

If one or more LMHs are detected by a SMH during the synchronization stage of the current service cycle, then the SMH is in the transmission range of the LMH. The SMH will sleep until the time designated for that LMH to begin its transmission. The SMH will listen to the index transmission of each LMH it detected. The SMH will also listen to the static data portion of the transmission during the first broadcast transmission it hears. A SMH will also be able to obtain any dynamic data items transmitted by any of the LMHs it detected. These items will be listed in the index. An index contains the name, or other identifier, for each item in the broadcast and they order in which they will be transmitted.

When a SMH is receiving an index or data item transmission, the SMH will be in doze mode. A SMH receiving a transmission, retrieving data items and storing them locally is said to be listening to that transmission. At all other times during data push, a SMH will be in sleep mode. In doze mode, a LMH/SMH is able to receive a transmission, retrieve the data items in the broadcast and store them locally.

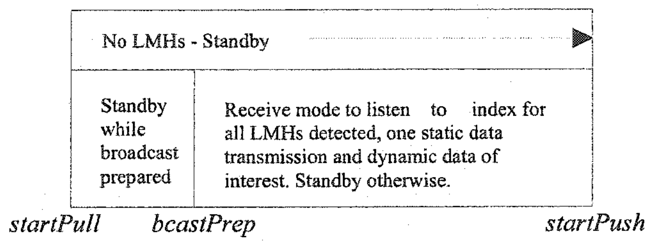


Figure 3-7 SMH Broadcast Stage

Whenever a SMH could be sleeping, it is also capable of remaining in doze mode or switching to active mode to perform other tasks. However, SMHs are prevented from transmitting during data push. Data queries and peer-to-peer messages may only be transmitted during the data pull stage that follows. The bandwidth of a MANET is small and needs to be reserved for the data broadcast.

The behavior of SMHs during the Data Push Stage is shown in Figure 3-8. Following the data pull stage is the data push, which is described next.

Every server SMH will run this algorithm

time = startPush
If (No LMHs detected) Standby
else
For (i = 1 to Num_{LMH})
SMHs who detected LMH_i during synchronization stage
(Standby when not doing one of these items / Receive otherwise)
1. Listen to index
2. Listen to static data one time
3. Listen to dynamic data of interest.

Figure 3-8 Algorithm for SMH Period of Data Push Stage

3.4 TriM Data Pull Stage

In the data push stage we have data broadcast. This is one of the three data communications methods of a MANET. Data broadcast occurs first, as data pull is not as energy-efficient as data broadcast. In the data broadcast stage the data needs of many clients (SMHs) can be satisfied with a single transmission. During the data push stage the limited bandwidth of wireless communication is restricted to the transmission of indices and data broadcasts. Data pull is delayed until after the conclusion of the data push stage.

During the data pull stage, this protocol allows the performance of the two remaining MANET data communication methods. These methods are data query and peer-to-peer communication. During data query, LMHs respond to data requests from SMHs. A LMH may also be asked to do routing of peer-to-peer communications.

A SMH is required to wait until the data pull stage begins to request data or communicate with other SMHs. This delay prevents SMHs from transmitting during data push, interfering with LMH broadcasts. This reserves the limited bandwidth of a wireless network for the data broadcast. This delay for data service has the effect of increasing response time for data not in the broadcast by delaying requests until the data pull stage. This delay allows the LMHs to have complete access to limited available bandwidth while transmitting broadcasting – a period of high-energy consumption for the servers. This delay is a tradeoff to protect the bandwidth needed in data broadcast. The benefit of this tradeoff is that all data broadcasts can occur without interference. As data broadcast has the potential of serving the maximum number of data needs at the lowest possible energy cost, preventing interference with the broadcast is important.

In data query, SMHs request data from LMHs when the data they need is not in the recent broadcast. In peer-to-peer communication, SMHs communicate directly with other SMHs. This direct communication between SMHs sets the MANET apart from other traditional wireless networks.

As a message may be intended for a LMH or a SMH, a message begins with ID of the message recipient. If the recipient is a LMH the message may be either a data query or a routing request. This message is sent to the nearest LMH, as determined by location information recorded during the synchronization stage. If the receiving SMH was detected during synchronization, a peer-to-peer message will be addressed directly to this SMH.

When a SMH wants to communicate with another SMH that is not within its direct transmission range, the SMH may communicate via the LMHs. Routing requests are sent to the nearest LMH detected during synchronization of the current service cycle. In this case, the LMHs are only providing routing services for the communication between the peers. If LMHs are needed for routing and no LMHs are within the transmission range of the transmitting SMH, the message cannot be delivered and the packet is dropped.

There are two different reasons a SMH may need to use a LMH during data pull. A SMH may need to make a data request and a SMH may need to use the routing services of a LMH. Because of this a control bit follows the node address at the beginning of the message. If the bit is set the message is a data query and the LMH will provide data service. If the bit is not set the message is a communication between SMHs. The ID of the destination SMH follows this bit. The LMH will route the message, without reading and processing the message further. The routing protocol used is independent of this data communication protocol.

Any SMH may transmit a data request or peer message or respond to a peer message. Any LMH may process a data request, route SMH messages when requested and communicate with other LMHs. However, all nodes are aware that their transmission may not be heard as nodes detected during synchronization may now be out of transmission range. For this reason, SMHs will not retransmit the same query or peer message during the data pull stage in a single service cycle. As all LMHs have a complete copy of the database, failure to receive a response to a data query will either be because the LMH has moved out of transmission range or the LMH did not have sufficient time to service the query.

The data pull stage has a set time, as determined at network deployment. Each node knows the time associated with data pull, which is of length *pullLen*. Each node tracks time as it passes, and moves into the appropriate stage at the indicated time.

3.4.1 LMHs – Data Pull Stage

The actions of active LMHs during the Data Pull stage of the service cycle are shown in Figure 3-9. A LMH is active if either other LMHs or SMHs were heard during data synchronization. If a LMH heard no LMHs or SMHs during synchronization, the tasks of data pull will not be required of that LMH. If this occurs, the node is considered inactive and enters standby mode during this stage.

The active LMHs have three tasks during the data pull stage. First they must respond to data queries (data-on-demand) from SMHs. Any data query that is not serviced during this data pull stage is added to the broadcast in the data push stage of the next service cycle. Second, LMHs must route SMH peer-to-peer messages when requested. Finally, any LMH to LMH communication, such as data synchronization, takes place during data pull.

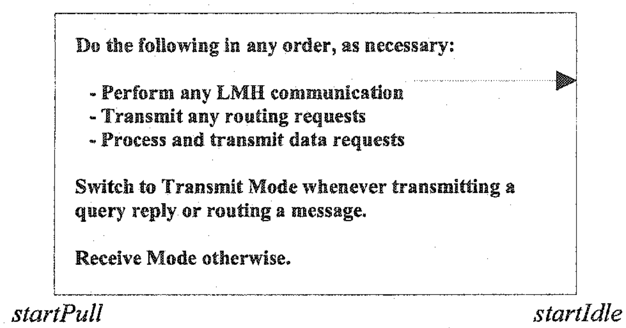


Figure 3-9 Active LMH Data Pull Stage

The work of a LMH during the data pull stage is wrapped up in these three tasks. It is always possible that no SMH will make a data or routing request and that no LMH communication is needed. In this case, the active LMHs will be in receive mode.

Throughout this stage a LMH remains in receive mode whenever no transmissions are pending. If a transmission becomes necessary, the LMH will switch to transmit mode long enough to make the required transmission, either responding to a data request, routing request, or a transmission from another LMH. Receive mode is a relatively inexpensive state for the LMH when compared to transmission.

If data queries are received by a LMH, the LMH will process the query, switch to transmit mode to transmit a response and then will return to receive mode. If multiple queries are received, the LMH will queue queries in the order received. If a SMH queries for a data item currently in the LMH's request queue, the LMH ignores the additional query. All requests are then satisfied upon the single transmission of the data item. At the end of the data pull stage any data queries remaining in the request queue are stored for addition to the next broadcast. These unserved data requests make up the dynamic portion of the next broadcast. They are called dynamic not because their contents change as the value of any data item may change over time. They are referred to as dynamic because the data items that make up the dynamic portion of a broadcast change from one service cycle to the next. The behavior of LMHs during the Data Pull Stage is shown in Figure 3-10.

Every server LMH will run this algorithm

time = startPull
If no LMH or SMH detected
    switch to standby
Otherwise (Active)
    switch to Receive Mode

repeat (if Active)

  if query is received and request queue is empty
    Process Query
    Switch to Transmit Mode
    Transmit Data
    Switch to Receive Mode

  else if query is received and it is in the current request queue
    Drop Query.

  else if query is received and it is NOT in the current request queue
    Add query to queue. Process in order received.

until time = startIdle

if at time = startIdle, unserved queries remaining the request queue, mark the data items requested for inclusion in the dynamic data portion of the broadcast transmitted in the next service cycle's data push stage.

Figure 3-10 Algorithm for LMH Period of Data Pull Stage

3.4.2 SMHs – Data Pull Stage

SMHs have a simpler situation than LMHs. A SMH has only a few potential situations as shown in Figure 3-11. If a SMH detected no LMHs and no SMHs during synchronization, it will be inactive during this stage and will switch to standby mode. Otherwise, the SMH is considered active.

If a SMH is active, the first situation is when a SMH needs to query for data not included in the broadcast. It is the responsibility of each SMH to manage its own data needs. The SMH will be in transmit mode while transmitting the data query and will be in receive mode as it awaits a response. A SMH is capable of listening to a data transmission, retrieving the data and storing it locally while in receive mode, however a SMH must be in transmit mode to transmit.

Second, a SMH may need to communicate directly with another SMH. If the target SMH was detected during the most recent synchronization stage, it will transmit to the target SMH directly. If the target SMH was not detected, the necessary control bit will be set and the message will be sent to the nearest LMH for routing service. During these transmission activities the SMH will be in transmit mode. When it concludes these activities, the SMH will change to receive mode.

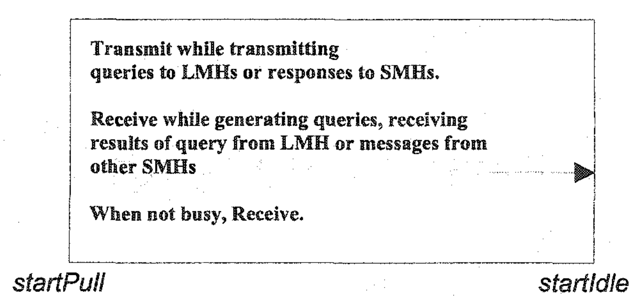


Figure 3-11 Active SMH Data Pull Stage

Finally, a SMH may receive a transmission from another SMH. If this occurs, the receiving SMH will switch to receive mode if it is not already and create a reply. The SMH will then switch to transmit mode in order to transmit the reply. When not transmitting during one of these three activities, an active SMH will be in receive mode.

Whenever a SMH wants to transmit, the SMH will listen to the channel and will transmit the packet when the channel is clear. A SMH will doze when it is not required to be in active mode. The behavior of SMHs during the data pull stage is shown in Figure 3-12.

Every client SMH will run this algorithm

time = startPull
If no SMHs and no LMHs detected during synchronization
switch to Standby Mode
Otherwise (Active)
switch to Receive Mode

Repeat (if Active) – Process in the order received, queuing up tasks as they arrive

If SMH has data needs
Prepare Query
Switch to Transmit Mode
Transmit query.
Switch to Receive Mode

If SMH receives a message from another SMH
Process Message
Switch to Transmit Mode
Transmit Response
Switch to Receive Mode

if SMH needs to communicate with another SMH switch to Transmit Mode
if destination SMH detected during Synchronization then Transmit message
else transmit message to a LMH for routing.

until time = startIdle
Clear task queue. Tasks do not persist from one service cycle to the next.

Figure 3-12 Algorithm for SMH Period of Data Pull Stage

SMHs will only be expected to handle one request at a time. As bandwidth is limited, only one transmission will be heard at a time. Later transmissions received while processing earlier transmissions will be queued until the current task is processed and transmitted. Any items remaining in the task queue at the end of the data pull stage are dropped.

3.5 TriM Idle Stage

Following the data pull stage; a MANET will enter into a period where all nodes sleep. The length of this period is determined by the network designer and is set at network deployment. The length of this idle period is kept by parameter idleLen. An idle period allows all nodes to switch to standby, increasing the length of time the network can remain active. Standby spreads out the data push and data pull stages and uses very little battery power. This period is determined by the necessary frequency of broadcasts for the network to perform its designed functions.

Following the idle stage, the service cycle will repeat, beginning in the synchronization stage.

3.6 TriM Conclusion

In Chapter 2 we discussed the issues surrounding MANET data communication protocol and discussed how current research has dealt with these issues. It was shown that current research, while advancing the state of the art in MANET data communication, still has many tasks that could be improved.

In this chapter, we have proposed TriM, a new MANET data communication protocol that considers the issues of mobility and battery power for both servers and clients while permitting all three forms of MANET data communication, data broadcast, data query and peer-to-peer communication.

In Chapter 4 we discuss the absence of a standard way to set-up and evaluate MANET data communication protocols. To address this, a benchmark for MANET data communication is proposed. This benchmark contains a standard architecture, workload and set of evaluation criteria.

Chapter 5 will provide an evaluation of the TriM protocol mathematically according to the proposed benchmark. Chapter 6 will contain the results from simulating TriM, using the proposed benchmark. The results of this evaluation will be compared to two recent MANET data communication protocols. The protocols chosen for evaluation are Wieselthier's [72] and Gruenwald's [37]. The protocols chosen for comparison provide the most information about the protocols involved, making it possible to make comparisons. In addition, both protocols are recent and representative of the protocols currently available.

Chapter 4

MANET DATA COMMUNICATION BENCHMARK

In this chapter we propose a benchmark for MANET data communication protocols. The reason this is even an issue is that no benchmarks exist for any MANET protocols, including data communication protocols. Because of this, each research study independently chooses the architecture, network workload parameters and evaluation criteria. The result is that comparing and evaluating proposed MANET data communication protocol becomes difficult, if not impossible.

We begin by reviewing the use and construction of benchmarks in computer science in the recent past, focusing on database system benchmarks. In Section 4.2, the current state of research is discussed as it relates to MANET architecture and use, network workload and protocol evaluation. This discussion centers on a representative group of seven MANET data communication protocols. In Section 4.3 the proposed benchmark is presented. This is followed by conclusions in Section 4.4.

4.1 Introduction to Benchmarks

Benchmarks are an accepted way to compare and evaluate different protocols, algorithms and architectures. The use of a benchmark indicates that an area of research is well established and that the associated research community collaborates [66]. When a benchmark is introduced into a research field, there are several potential benefits. Sim, et. al. suggest that the introduction of a benchmark into any field signifies an increased level of discipline maturity, increased technical progress, and greater collaboration and community among the area's researchers

[66]. The development of a widely accepted benchmark is itself an act of collaboration.

Benchmarks can be simple or extensive. There is value in being able to compare computer systems, protocols and algorithms according to a standard, non-vendor specific benchmark. Benchmarks increase the ability to make meaningful comparisons.

An example of a simple and informal benchmark is the common practice in computer vision research to use the same images that previous algorithms have used. The point of using a common set of images is to allow your algorithm to be compared with previous algorithms on the same data. Knowing that your algorithm does well on *Image X* when everyone else is using *Image Z* may not be useful unless you also process *Image* Z. This practice is sufficiently common that the Computer Vision Home Page archives the location of many of the shared images used by computer vision researchers [11].

While informal benchmarks exist, more formalized benchmarks are widespread. Benchmarks occur in all areas of computer science from programming languages to architecture.

For example, benchmarking has been used to demonstrate the ability of Java to compete with Fortran and C in scientific applications. There has been concern that Java is not an appropriate language for large scale and parallel scientific applications. The general feeling has been that Java could not perform at the same level as these more traditional scientific programming languages. The Java Grande Benchmark Suite was implemented for Java, C and Fortran on a number of architectures. The results demonstrated that the performance difference between

Java, C and Fortran has become much smaller [23]. These results demonstrated, perhaps convincingly, that Java is an acceptable choice for these types of applications.

A common theme in computer architecture development is the increase in speed with each successive generation. However, when considering the vast number of architecture choices, it may be difficult to choose one over another. Examples of hardware benchmarks are those provided by the Standard Performance Corporation (SPEC) [10]. SPEC has a large variety of benchmarks. These include SPEC CPU 2000 that measures the performance of CPUs, memory and compilers and SPECweb99 that measures the performance of W.W.W. servers [10].

Even within specialized architectures a benchmark can be helpful. For example, when the National Center for Atmospheric Research (NCAR) put out a competitive bid for the purchase of one or more supercomputers, they also provided the criteria for selection in the form of a benchmark [40]. The NCAR benchmark measured the type of workload they anticipated for the purchased supercomputer(s) [40].

Within the database community, benchmarks have also been common with both general purpose and specialized benchmarks in use. Over 20 years ago the database group at the University of Wisconsin-Madison was working on database benchmarks. Their goal was to develop a way to measure the performance increase that could be had on existing architectures through the use of specialized hardware and software [20]. The focus here was not on the underlying machine but in ways the underlying architecture could be utilized to improve database performance. They looked at both the performance of specialized database hardware as well as the performance of DBMS software on non-specialized architectures [20]. This benchmark was developed with a specific set of database queries that tested the range of available operations and a database that was highly tuned for this set of queries [20].

More recently, The Transaction Processing Performance Council (TPC) [13] provided a common database benchmark. TPC provides the TPC-C benchmark to measure the performance of transaction processing and database systems [13]. The TPC-C benchmark is based on a banking debit/credit model and measures transactions per second [13].

However, one benchmark does not suit all needs. As databases are used for an ever-wider selection of applications, new benchmarks become necessary. For example using TPC-C to evaluate a database developed for image storage and retrieval may not provide meaningful results. Image storage and retrieval are different from a debit/credit situation. When image retrieval is based on image content, as occurs in Query By Example (QBE) a new benchmark is needed to compare systems. In a Computer Base Image Retrieval (CBIR) system three specific parts are necessary. These are the image database, judgment on the relevance of images retrieved in QBE or other methods, and a measure of the performance of the system doing image retrieval [59]. These are not parts of the TPC-C benchmark.

The need for a new benchmark may not be because of a new application area. Instead it can be necessitated by a change in the paradigm used. For instance, the relational model has dominated in database systems for many years. New paradigms such as the object-oriented model in database systems are emerging. A benchmark that measures the performance of a relational database may not be capable of providing meaningful comparisons of other database model.

As early as 1988, comparisons were being made between relational and object-oriented systems [33]. The Sun Benchmark was developed in an attempt to make more meaningful comparisons between relational and object-oriented systems [33]. These types of comparisons between models are common when new models emerge, much like the previously discussed comparison between Java, C, and Fortran for scientific computing. However, at some point the benchmark proves inadequate.

As object-oriented database systems have become more common, benchmarks have been built specifically to measure the performance of Object-Oriented Database OODB systems to allow comparison and evaluation. One recent OODB benchmark is oo7. This benchmark was developed at the University of Wisconsin-Madison, to measure the performance of OODB systems [24]. This benchmark is well suited to test the overall performance of an OODBMS running a Computer Aided Design (CAD), Computer Aided Manufacturing (CAM) and Computer Aided Software Engineering (CASE) application [29].

If measuring the general overall performance of an OODBMS, this or similar benchmarks would be sufficient. However, this benchmark may not be sufficient to measure the performance of specific targeted protocols in an OODB system. One such protocol is the clustering protocol proposed by Darmont, et. al. [29]. Darmont wished to measure the performance of a variety of clustering protocols to compare against the protocol he proposed. The general performance of the OODBMS was not the issue. Instead, the more specific issue of the performance of the clustering protocol needed testing. For this purpose, a more precise benchmark was developed by Darmont to allow the comparison of clustering protocols in OODBMS [29].

Mobile computing is no different and has at least one benchmark. As mobile computing is less mature than relational and object-oriented database research, its benchmarks are of more recent design. One benchmark, perhaps the first in mobile databases, was developed by Seydim and Dunham [65]. The purpose of their benchmark was to measure the performance of location dependent queries in traditional mobile networks [65]. A location dependent query is one whose value depends on the location of the client. For example, a query asking for nearby hotels is very much constrained by your location.

Compared to other research areas, MANET research is not very mature. Benchmarks must still be developed, as there is currently no MANET benchmark of any type in the literature. There is no standard architecture for Mobile Ad-Hoc Network data communication protocol testing and there is no standard set of evaluation criteria designed to measure the effectiveness of MANET data communication protocols. Both of these issues can be addressed through the careful creation of a benchmark for MANET data communication protocol evaluation. One result of this research is the development and proposal of a standard benchmark for MANET data communication protocols.

In the case of this research, a standard architecture and evaluation criteria is needed to measure the performance of MANET data communication protocols and to measure this performance against the performance of other MANET data communication protocols, such as those proposed by Wieselthier [71] and Gruenwald [37].

Like the NCAR benchmark, care must be taken to develop benchmarks that serve their constituencies. The NCAR benchmark served to eliminate from the bid process supercomputers whose performance was less than the NCAR was willing to consider. According to Kumar, et. al, a benchmark should consider all constituent groups including the purchaser, hardware developers, and where appropriate, software developers [50]. The benchmarks developed should review a wide range of existing systems, consider the use of the systems, focus on functionality and not implementation, and be publicly available [26].

A benchmark developed for data communication protocols needs several parts. Darmont suggests that an OODB clustering benchmark requires a defined database and workload [29]. Seydim suggests that a benchmark to measure the performance of location dependent queries needs three things: data, queries, and execution guidelines [65]. The point is that a benchmark must have several parts and those parts must be what is required to do the desired comparisons.

The benchmark's previously discussed all use some data to load the systems, select a workload and provide evaluation criteria. There are a variety of ways in which these items is created. Seydim [65] suggests that the locations for restaurants and hotels could be obtained from a real database. However, Seydim generates this data randomly using a Gaussian distribution [65]. Once the types of queries needed for testing were determined, the benchmark queries were created randomly [65]. The evaluation criteria selected were the criteria important to Seydim [65]. To develop the workload for the OCB protocol, Darmont determined the types of queries that would allow evaluation of OODBMS clustering and then selected a set of these queries to form the basis of his protocol [29]. In the Darmont protocol, the evaluation criteria selected were those that show the effect of the OCB protocol on clustering in OODBMS [29]. The design on the oo7 protocol is also instructive. A range of values were used for several parameters were used to handle a variety of situations [24]. For instance a small, medium and large database size were specified. No justification for the sizes selected is given. These numbers appear to be based on previous benchmarks, current databases in use and experience. The Sun Benchmark uses a similar approach. This benchmark was developed to compare the response time to seven fundamental database operations using a simple schema [33]. Two database sizes were also used, although a rationale for the sizes was not provided. A large database was ten times larger than the small database [33].

The MANET data communication benchmark will have three parts. These are:

- ***A standard architecture***. Before comparing two protocols, we need to know they are running in similar situations. A battlefield MANET has different characteristics from a business meeting scenario – beyond the presence of enemy fire.
- ***A workload***. The workload is a description of what the network is doing with respect to data communication. This workload will include specification for the size of the database and other parameters associated with data communication.
- ***Evaluation Criteria***. Once a MANET is built, deployed, and running, a way to compare performance of different data communication protocols is provided. This evaluation will use architecture, workload, and protocol specific values.

The remainder of this chapter is organized as follows. Section 3.2 discusses the previous work in MANET routing and data communication as it relates to the three parts of the proposed MANET data communication benchmark. This discussion will be in three parts. First, architecture issues are discussed followed by workload issues. This is followed by a discussion of typical evaluation measures.

In Section 4.3 the proposed MANET data communication benchmark is presented. This also occurs in three parts. First we address the benchmark's architecture. This is followed with the workload discussion. Section 4.3 concludes with the evaluation criteria selected. The chapter ends with a summary of the proposed benchmark.

4.2 Background for Proposed MANET Data Communication Benchmark

In this section we discuss the current state of MANET data communication protocol research. The discussion focuses on the MANET architectures selected, the uses envisioned, the network workload chosen, and the evaluation criteria used. This discussion will then lead to the proposed MANET data communication benchmark.

4.2.1 MANET Routing and Data Communication Architecture Background

IEEE 802.11 specifies the medium access control (MAC) and physical layer for wireless networks [2]. This standard defines an ad hoc network as:

> "A network composed solely of stations within mutual communication range of each other via the wireless medium (WM). An ad hoc network is typically created in a spontaneous manner. The principal distinguishing characteristic of an ad hoc network is its limited temporal and spatial extent. These limitations allow the act of creating and dissolving the ad hoc network to be sufficiently straightforward and convenient so as to be achievable by non-technical users of the network facilities; i.e., no specialized "technical skills" are required and little or no investment of time or additional resources is required beyond the stations that are to participate in the ad hoc network." *[2]*

The standard points to the temporary, autonomous, and mobile nature of these networks. Among the characteristics of mobile nodes pointed out in 802.11 are:

- Nodes may transmit while in motion.
- Battery powered nodes may not be turned on at all times.
- Networks lack full connectivity.
- Networks have asymmetric propagation.
- Network topology is not pre-planned.

The standard addresses the various concerns at the MAC and Physical Layers, including communication between service areas, data security, frame construction and addressing, as well as tracking nodes moving in and out of a service area.

However, the arrangements of specific MANET configurations are not discussed. This is appropriate. The standard is not for one particular type of wireless networks and must provide guidance to the entire spectrum of wireless networks. The standard is valuable in that it discusses the differences between wired and wireless networks as well as providing a framework for medium access control and physical layer design.

A number of research results in MANET data communication have been published which provide information on network configuration. Tables 4-1 and 4-2 show some common network set-ups. When the information was not given in the paper referenced, the entry is marked unknown.

These seven studies have been selected, as they are all MANET data communication protocols. While six of the data communication protocols only include data broadcast, [37] also includes a form of data query. The remarkable thing is that while each protocol concerns MANET data communication, there is a large degree of variation between them. While these protocols may have been testing different application scenarios, this is not mentioned in these reports even when example scenarios are included in the introduction. The results of these studies are difficult to compare due to the variation of architectures, workloads, and evaluation criteria.

	#nodes	mobility (m/sec)	Region Size (m)	Simulation Time (sec)
Gruenwald [37]	3-5 LMH 1000 SMH	unknown	500 x 500	1000 transactions
Jung [46]	256	unknown	200 x 200	unknown
Kunz [51]	50	1 – 20	1K x 1K	900
Tang [68]	75 - 300	unknown	500 x 500	120
Tseng [69]	100	0 – 30.5	500 x 500 to 5500 x 5500	unknown
Wieselthier [70][71][72]	50	unknown	5 x 5 Grid	1000 work units
Williams [73]	20 – 110	unknown	350 x 350	100
	60	unknown	350 x 350	100
	60	1 – 20	350 x 350	100

Table 4-1 Hardware Configuration of Several MANET Research projects

In Table 4-1, the number of nodes, amount of node mobility, size of simulated regions and length of simulation are given. When the research paper differentiates between servers and clients, this is indicated. LMH are the servers and SMH are the clients. In the Williams study, simulations were run for all three sets of values. When a range is given, that is the range used by the research.

In addition to the number of nodes, amount of mobility, and area of network deployment we need to additional information about the capabilities of network nodes. Characteristics such as node bandwidth, CPU performance and power dissipation rates are all important when considering the affects of data communication on an ad-hoc network.

Table 4-1 shows some agreement over network size, number of nodes, and transmission range of nodes. There is also a general agreement that node bandwidth is limited. Studies that include it generally place bandwidth at around 1 Mbps. For other parameters, there is no agreement. For instance, there is a great deal of variation in the amount of mobility that should be allowed.

In addition, most simulations are so short that the real behavior of the networks with respect to data service and power dissipation may not be visible. Simulations should cover a time period more representative of the network's use. No one should suggest that simulating a data communication protocol during the first 100 seconds of a MANET deployment (just over 1.5 minutes) gives an accurate picture of likely performance of that protocol.

Another problem is immediately clear. Not every study contains sufficient data to compare even this small number of parameters. It is essentially impossible to adequately compare the performance of different data communication protocols.

The parameters given in Table 4-1 do not fully parameterize the unique and defining characteristics of a MANET and these parameters are insufficient. A MANET is characterized by low bandwidth and the need to conserve power. In addition, nodes acting as data servers typically have more power and better storage and computation resources. Because of this, the power and bandwidth characteristics of both servers and clients need to be considered.

In [37], some additional parameters are provided. These are listed in Table 4-2. With the exception of [37], the studies discussed do not include parameters for CPU processing power or for power dissipation rates. This suggests that these studies may not have considered these parameters. When considering data communication protocol in a MANET deployment, the bandwidth and power of node CPUs is important information. In addition, power dissipation rates allow the network designer to predict the amount of network disconnection due to battery failure during network deployment.

In addition to the characteristics listed in Table 4-1, the additional parameters such as those in Table 4-2 should be included in any MANET architecture benchmark. As clients may query the data servers and may communicate directly, the number of clients and the effect of their mobility to data access patterns are important to consider in data communication protocols.

Parameter Description	Default Value
CPU Power	LMH – 140 MIPS SMH – 4 MIPS
Power Dissipation Rate	LMH – 170 W/hr SMH – 7 W/hr

Table 4-2 Additional MANET Parameters [37]

The real problem appears that studies, even when they list them, do not directly consider the type of environments in which the MANET will be used. Some scenarios have been suggested. These include a military battlefield, domestic rescue operations and temporary business networks – such as a trade show in a non-traditional or unwired environment.

Considering the scenario of a MANET is important. The characteristics of each scenario directly impact the design of the network. These scenarios have similarities and differences. The three main scenarios suggested for MANET deployment are discussed in Sections 4.2.1.1 – 4.2.1.3.

4.2.1.1 Battlefield Scenario

In a battlefield situation, the nodes are typically carried either on vehicles on by soldiers. A vehicle such as a tank or humvee can easily carry the larger and more powerful server. These vehicles may move very rapidly. Soldiers, carrying the lighter clients, will at times be transported by vehicle and at times will be on foot. Vehicles and soldiers can have vastly different rates of travel, but typically move in the same general direction. The relative movement between nodes may vary greatly.

In this scenario, data delivery is time critical. The data delivery may require a firm or soft real-time solution. In a firm real-time network missing a deadline is a network failure. Soft real-time systems must generally deliver data within the deadline but periodic failures to deliver data on time do not constitute network failure. The geographic area covered in this scenario can be quite large and networks may be sparsely populated with frequent partitions. A firm real-time MANET may not be practical.

The MANET can best be deployed at the brigade level, where all nodes are mobile [64]. A forward brigade has as many as 1000 SMHs with 15 to 20 LMHs covering an area roughly 10 by 15 kilometers [64]. Above the brigade level, portions of the network are not wireless. While a brigade is deployed as a group from a staging area, movement during battle, both direction and speed, is random.

4.2.1.2 Domestic Rescue Scenario

In a rescue situation, vehicle movement is typically slow and restricted. Much of this vehicular travel will follow existing roads. Roads are generally grid-like with roads occurring every 70 to 150 meters [46]. Much travel is on foot, leading to regular but slow topology changes. While the direction of travel may be highly unpredictable, the region of travel is not. All movement would be confined to a set geographic region.

It is important to consider the size of the rescue team and the types of rescues typically performed. A Federal Emergency Management Agency (FEMA) Incident Response Team is made up of 40 individuals, providing 24 hour incident support [5]. The size of the team that may be involved in a rescue involve small teams of 3 to 5 up to large teams of 50 to 60 [4]. The types of rescues are usually building collapses due to earthquake and terrorist activity. Examples of recent rescue attempts are earthquakes in the Philippines and Armenia and the terrorist activities in Oklahoma City and New York City [4][9]. It is important to note that in each of these cases, the area of concern was relatively small and static. Radio range was not an issue.

From this we see that data delivery is important, and some types of information may be time-critical. The network may be large, but disaster areas are generally broken into smaller work regions. Each region could have its own MANET. In this type of applications, the number of clients served by each server is not very large. Firm or soft real-time systems may be practical in these situations.

4.2.1.3 Business Scenario

A temporary network for a trade-show or similar business situation has another set of characteristics. There will be little or no vehicular traffic. Rather clients will be traveling with individuals on foot. Servers may be mobile, or may be stationary. This is still a MANET as the servers are battery powered and can be moved, even if they are not moved. The area of interest will be easily defined and battery-powered servers could be placed to provide full coverage. Data delivery may be important, but is probably not time-critical. This may be the smallest network geographically, while having the largest number of nodes.

Two examples of the type of spaces used for conventions are the Odeum Sports and Expo center near Chicago, Illinois and the Dallas Convention Center in Dallas, Texas. The Dallas Convention Center has a series of connected exhibit halls that represent 800,000 square feet. This area is roughly L-shaped, in a bounding box of approximately 600 x 1800 feet. A smaller single exhibit hall is 330 x 590 feet, containing 225,000 square feet [3]. The ODEUM Sports and Expo Center has adjacent exhibit halls of approximately 20,000 square feet each. The halls are approximately 100 x 200 feet [8].

A MANET server with a range of 250 – 350 meters can easily handle the smaller halls. The larger area of the Dallas Convention Center would need a few servers, but could easily be handled with 3 or 4.

Within this environment, there are potentially hundreds of clients. These clients would rarely if ever become disconnected from the server because of location or mobility. These clients would become disconnected if they were powered off or their batteries became depleted. In addition, keeping servers databases fully replicated would be a much simpler task, as the servers would be able to stay within range of the other servers.

This scenario considers a large meeting or trade show. Within this environment, individuals carry the clients as they walk within the region. Johannsson, et. al. [43] refer to this as the event coverage scenario. In their model, 50 highly mobile users are simulated [43]. They suggest that typical mobility will be walking and can be expected in the to 0 to 1 m/s range [43].

In a large meeting, there could be several hundred participants. There may be as many as 1000 participants, although this would be at the high end of estimates. Each participant may be in possession of a client node. If all clients were to query the database at the same time, the bandwidth of all nodes could be easily saturated. Certainly the convention or other business meeting would not need to take place indoors. However, areas of similar dimensions are anticipated.

The preceding sections have discussed the definition of a MANET according to IEEE 802.11, the architecture used in a variety of MANET research projects and the type of scenario in which a MANET is anticipated to be of use. This leads directly to a proposed architecture for the MANET data communication benchmark, as defined in Section 3.3.1.

4.2.2 MANET Data Communication Workload Background

Within the workload portion of the MANET data communication protocol, we consider the database and communication needs of the network. It is assumed that a routing protocol is available for use by the data communication protocol. There are several to choose, such as Aggelou's RSMAR [16], Haas' ZRP [39] or Ko's LAR [48]. MANET routing protocols are briefly discussed in Chapter 2. The benchmark states no routing protocol preference. The benchmark assumes an appropriate routing protocol exists and seeks to be routing protocol independent. Table 4-3 provides a comparison of workload parameters for the same protocols compared in Table 4-1. These are all MANET data communication protocols.

The three lines in Table 4-3 for the Williams entry correspond to the three sets of values shown for Williams in Table 4-1. Table 4-3 shows little agreement on the parameters listed. In addition to these parameters a data communication benchmark providing data broadcast needs to specify the size of the database used. There are no scenario specific workload needs. This discussion leads to the workload of the proposed MANET data communication benchmark, discussed in Section 4.3.2.

	Packet Frequency (pkt/sec)	Packet Size (byte)	Broadcast Radius (m or unit)
Gruenwald [37]	unknown	25K	LMH – 200 SMH – 100
Jung [46]	unknown	120	unknown
Kunz [51]	4	512	250
Tang [68]	unknown	unknown	unknown
Tseng [69]	unknown	280	500
Wieselthier [70][71][72]	unknown	unknown	unknown
Williams [73]	10 1 – 80 10	64	100

Table 4-3 Workload Parameters in Current
MANET Data Communication Research Studies

4.2.3 MANET Routing and Data Communication Evaluation Background

In the following sections we discuss the evaluation criteria used by the proposed MANET data communication benchmark. There are a number of measurements used to evaluate the performance of current MANET routing and data communication protocols. While most of the criteria examined are for routing protocols, routing and data communication have some similar issues. These criteria give us a starting point for proposing evaluation criteria for the proposed MANET data communication benchmark.

4.2.3.1 Access Time (Tuning Time)

This is a measure of the effectiveness of the broadcast portion of the data communication. Access time is a measure of the length of time a client must actively listen to a broadcast [38]. In a MANET this is important because it takes a client power to actively listen to a broadcast. In [25] this measure is called Average Broadcast Length (ABL). The average length of the broadcast is directly related to the average access time.

4.2.3.2 Broadcast Effectiveness

This measure is an average of the ratio of packets received to packets sent or the ratio of the number of nodes reached to the number of nodes a broadcast was supposed to reach.

In [72] this is given as:

$$e = \frac{1}{X}\sum_{i=1}^{X}\frac{m_i}{n_i} \tag{4-1}$$

X = number of multicasts,
m_i = number of nodes reached by multicast i,
n_i = number of nodes multicast i attempted to reach.
i is an index for each multicast

Broadcast effectiveness is used by several other research studies, but often has a different name. Broadcast effectiveness is used by [76] (called success ratio), [30] (called fraction of packets delivered), [51] (called effectiveness), [73] (called delivery ratio), [61] (called packet loss), [34] (called robustness), and [43] (called percent of packets received).

4.2.3.3 Hit Ratio

While broadcast effectiveness measures the number of nodes reached or the number of packets dropped, hit ratio measures the usefulness of the packets that arrive. Simply put, hit ratio is a measure of how well the broadcast satisfies the need of clients [37]. If a hit ratio is high, clients have to request few items through data pull. In [38], a similar measure is called access probability.

Hit ratio, as given is for each individual broadcast. The formula for hit ration is:

$$p = \frac{I}{M} \tag{4-2}$$

I = Total requests/accesses for data item I,
M = Total requests/accesses for data items.

4.2.3.4 Number of Hops

Number of hops is a measure of the number of nodes between a data source and the target node [26]. In the case of our MANET architecture, this would be a count of the number of LMHs involved in routing a peer-to-peer communication.

4.2.3.5 Packet Delay

As described in [43], packet delay or end-to-end delay is the amount of time for a message to be delivered from the source to the destination. This is also a measure used by Williams and Camp to evaluate broadcasting techniques [73]. Chin also uses a packet delay measurement, calling it end-to-end latency [26]. Routing delay is also the measure in [43].

Packet delay may not be the best measure for overall MANET data communication evaluation, as routing is limited to peer-to-peer communication. However, as peer-to-peer is on of the data communication methods in a MANET, packet delay does have its place. As the majority of MANET work has been in routing, it is not unexpected that routing delays are a typical and common measurement.

4.2.3.6 Energy Consumed

While the energy consumed in a MANET is critical due to the battery-powered nature of the clients and servers, few studies consider energy consumption. Power consumption was studied in [49]. This was done to study the effect of requiring the server to sleep periodically. They found that periodic naps, even short ones, greatly reduced power consumption.

In studies proposing protocols for MANET broadcast, only a few consider power. In [68] power consumption consists of three parts, E_{Tx}, E_{Rx} and E'. These are the costs associated with transmitting and receiving a single packet. E_{Tx} is the cost to transmit a single packet while E_{Rx} is the cost to receive a packet. E' is an overhead cost associated with Media Access Control (MAC) for both sender and receiver. The equation then is:

$$W = E_{Tx} + E_{Rx} + E' \qquad (4\text{-}3)$$

E_{Tx} = Power required to transmit one packet,
E_{Rx} = Power required to receive one packet,
E' = MAC overhead.

In [37], power consumption is divided into two categories. Power consumed by clients and power consumed by servers is maintained independently. As the power available to these different nodes differs as does the tasks assigned, this is an important division.

Wieselthier, et. al. [72] take a different approach. Rather than track the power consumed, they try to measure the work accomplished by each unit of power. Wieselthier never actually use specific units of measurement. Instead they treat everything as a generic "unit".

The measure used by Wieselthier is the yardstick. The equation for the yardstick is given as:

$$y_i = \left(\frac{m_i}{p_i}\right)\left(\frac{m_i}{n_i}\right) \tag{4-4}$$

m_i = number of nodes broadcast i actually reached,
n_i = number of client nodes broadcast i attempted to reach.
p_i = sum of transmitter power used by all servers during broadcast i,
i = ith broadcast.

The yardstick measures the number of nodes reached for every unit of energy expended [72]. If we look at the two components of the yardstick in Eqn 3-4, we see the first term is a ratio representing the number of nodes reached per unit of energy consumed. The higher this number, the larger the number of nodes reached for each unit of power. For instance, if $m_i / p_i = 2$, then two client nodes are reached for each unit of power.

The second term is a measure of efficiency. The goal is for m / n to be equal to one. A result of one for the second term would mean that you reached every node you attempted to reach. Anything less than one for the second term of the yardstick indicates the percentage of client nodes we intended to reach.

The global yardstick is [72]:

$$Y = \frac{1}{X}\sum_{i=1}^{X} y_i \tag{4-5}$$

X = number of multicasts.

The global yardstick gives us an average of the yardstick all multicasts. The global yardstick measures the average number of destinations reached with each multicast.

4.2.3.7 Average Response Time

This is the most common measure of data pull performance. Average response time is the measure of the time it takes for a query to be serviced [38][77]. During this time, a client must actively listen. Shorter response times minimize the power consumed by the client.

Like other measures, this metric goes by many names. Among these are average time to serve [25], turn around time [27], and access time [37]. In [14], both average and maximum response times are captured.

Energy consumption is easily extended to data push by [37] as power consumption is measured by the time spent in certain activities (broadcast, request transmission, active listening, etc.). The same can be done for peer-to-peer activities.

4.3 Proposed MANET Data Communication Benchmark

We have now discussed the current state of MANET data communication protocol research as it relates to architecture, workload and evaluation. We are now ready to propose a standard benchmark for the comparison and evaluation of

MANET data communication protocols

In Section 4.3.1 a common architecture is proposed. Some of the architecture parameters are used in protocol evaluation equations. These parameters will include a variable name to be used in these equations. In Section 4.3.2, the workload is proposed. Again, some of these parameters are used in protocol evaluation, and a variable name will be included. Finally, in Section 4.3.3, the evaluation criteria are presented. Three types of values are used in the evaluation of MANET data communication protocols. These are:

- **Benchmark Parameters** – These are values specified in the architecture and workload of the benchmark.
- **Protocol Parameters** – These are values that are specified by the protocol or are determined by the behavior of the protocol. These parameters are described, but no value can be given.
- **Evaluation Parameters** – These are the values we are calculating in order to compare and evaluate different MANET data communication protocols.

In the evaluation criteria descriptions, these different parameter types are clearly stated.

4.3.1 Proposed MANET Data Communication Benchmark Architecture

The proposed architecture for the proposed benchmark is composed of two parts. The first part consists of the common elements found in all MANET scenarios. The second part of the architecture benchmark is the scenario specific portion. There are three scenario specific portions to the benchmark. Before discussing the scenario specific values, common values are presented first.

The general characteristics of MANET architecture were chosen after considering IEEE 802.11, current research in MANET data communication and scenario specific characteristics. Where there is no general agreement on the exact value, a value was selected using our best judgment. The values for CPU processing power and power dissipation rates reflect typical technology in MANET hardware, a LMH using a Intel Pentium IV 1.5 GHz CPU and a SMH using a Pentium III 450 MHz CPU [18][21]. As newer CPUs are constantly being introduced to the marketplace, adjustments will need to be made periodically to the benchmark.

Specific scenario values are based on the papers referenced with each scenario description above. When a better value is determined, the benchmark can be updated to reflect these improved parameter values.

While different parameters are possible, by adhering to a standard set of parameters comparison between methods becomes possible. However, in addition to testing a new protocol according to this baseline, parameters that vary should be varied to note the behavior of the protocol under the range of potential values.

There is nothing in a standard set of parameters that prevents a new data communication protocol from using additional parameters. While these additional parameters may serve little purpose when comparing protocols they may help describe the behavior of a specific protocol or have other uses, as determined by the protocol designers. For instance, an additional parameter representing a new application area or scenario would be appropriate. As these new scenarios become recognized, parameters may be standardized for them as well as part of an updated benchmark.

4.3.1.1 Common Benchmark Architecture Parameters

The common benchmark parameters were selected based on the parameters indicated in Table 4-2, with additional parameters suggested by [18][52b] and [37]. Values have been updated for the processors listed in Section 4.3.1. The selected parameters are shown in Table 4-4.

Parameter	Value	***Variable (if used in evaluation)***
Bandwidth LMH SMH	 2 Mbps 100 Kbps	
Communication Radius LMH SMH	 250 meters 100 meters	
CPU Power LMH SMH	 1700 MIPS 100 MIPS	
LMH Power Dissipation Rate Transmit Mode Receive Mode Standby Mode	 170 w/hr 20 w/hr 2 w/hr	 *transRateL* *recRateL* *stbyRateL*
SMH Power Dissipation Rate Transmit Mode Receive Mode Standby Mode	 7 w/hr 1 w/hr 0.1 w/hr	 *transRateS* *recRateS* *stbyRateS*
Simulation Time	1 hour (3600 sec)	*period*

Table 4-4 Common Architecture Parameters for MANET Data Communication Benchmark

As a proposed benchmark, it is anticipated that improvements and adjustments will be made over time as collaboration occurs. It is noted that power dissipation rates and the length of the simulation are used in the calculation of the evaluation criteria. Parameters that are not used in this manner still direct the operation and function of the network. This will have an effect on protocol performance. The following section will address the scenario specific benchmark parameters.

4.3.1.2 Scenario Specific Benchmark Architecture Parameters

It is impossible to have a single benchmark that encompasses all potential MANET scenarios. This benchmark deals with that situation by having a set of parameters that are specific to each scenario previously described. As new applications for MANET deployment are developed, parameters specific to the new scenario can be added to the benchmark.

Table 4-5 lists the parameters for the three scenarios described. Battlefield parameters are for a brigade deployment [64]. The number of each node type is used in the calculation of evaluation criteria. The other values explain the operation of the network. For example, mobility is the range of speed allowed in each scenario while size of region describes the geographic area covered by the MANET.

Parameter	**Battlefield Scenario** [64]	**Rescue Scenario** [4][5][9]	**Business Scenario** [3][8][43]	***Variable***
Number of Nodes LMH SMH	 20 1000	 1 to 10 10 to 50	 4 to 6 1000	 *numLMH* *numSMH*
Mobility – all nodes	0 to 20 m/sec	0 to 10 m/sec	0 to 1 m/sec	
Size of roaming region	10 km x 15 km	5 km x 5 km	1 km x 1 km	

Table 4-5 Specific Architecture Parameters for MANET Data Communication Benchmark

The nodes specified in Table 4-5 must be initialized as part of each MANET deployment. The rescue and business scenarios assume that the LMHs and SMHs are initialized in random locations throughout the roaming region. In the case of battlefield scenario, the nodes are initialized in a smaller region, to represent the staging of troops prior to battle. The LMHs and SMHs are deployed randomly along a 1000 meter line on the edge of the roaming region. Once the network is initialized, motion of all nodes is random with respect to speed and direction. However, the random speed must be within the mobility range specified for each scenario. The distance of travel is calculated by multiplying a random speed by the time elapsed since the previous location calculation. Each time distance traveled is calculated, a new random speed and direction is used. When the distance and direction of travel would take a node outside the roaming region, the node will travel only as far as the edge of the roaming region.

Upon inspection, these different scenarios represent very different situations. The battlefield situation involves a large number of high mobility nodes spread over a large geographic area. The rescue scenario represents a medium sized geographic area with less mobility and fewer nodes. Finally, the business situation models a small geographic area with few servers but many low mobility clients.

4.3.2 Proposed MANET Data Communication Benchmark Workload

No specific data request frequency, message frequency, broadcast size or database size is selected in the MANET data communication protocol. Instead a set of parameters is selected for these items used in the benchmark. This allows the testing of a wide range of potential situations. The selected values are shown in Table 4-6.

Data request frequency and data request size values are for data queries. The size of the LMH query response is the same as the size of one data item in a broadcast. Message frequency and message size are used in peer-to-peer communication. The values used in data broadcast are size of broadcast, data item size and index item size. The benchmark assumes that all SMHs have the same data request and message request frequencies. It is further assumed that the database is fully replicated among all LMHs.

Parameter	**Value**	***Variable***
Database Size	500/2000/5000 items	
Broadcast Size	50 / 100 / 200 items	*itemsBcast*
Index Item Size	128 bytes	
Data Item Size	64Kbyte	
Data Query Frequency	5/20/40 requests/sec	*reqFreq*
Data Query Size	256 bytes	
Message Request Frequency	5/20/40 messages/sec	*peerFreq*
Message Size	512 bytes	

Table 4-6 Workload Parameters for MANET Data Communication Benchmark

The following section will address evaluation parameters selected for the MANET data communication benchmark.

4.3.3 Proposed MANET Data Communication Benchmark Evaluation Criteria

The evaluation of a MANET data communication protocol is a complex matter. Two items require measurement. First, the ability of the proposed protocol to perform data communication must be evaluated. These data communication methods are data broadcast, data query and peer messaging. If the protocol cannot perform these functions, no further evaluation is needed. We do this by measuring overall data communication performance and by measuring the performance of each data communication method.

The characteristics of a MANET affected by or affecting data communication also need to be evaluated. These critical factors are mobility and battery power. If nodes cannot find and communicate with each other or nodes run prematurely short on power, the MANET protocol is usable.

We use three types of values in the evaluation portion of the benchmark. Some of the architecture and workload parameters are used. These parameters were given variable names in Tables 4-4, 4-5 and 4-6.

The MANET data communication protocol being evaluated has parameters associated with it as well. These are parameters that are dependent on the protocol for their value. These parameters are described, but their value cannot be given as part of the benchmark.

Finally, we have the evaluation parameters. The evaluation parameters are the values calculated during the evaluation portion of the benchmark.

In this benchmark, we group together a small set of criteria that measure the performance of a MANET data communication protocol. These evaluation criteria measure both general MANET behavior as well as communication mode specific performance.

The purpose of this benchmark is to codify a group of measurements that can be used as an evaluation suite in the evaluation of MANET data broadcasting protocols. Existing evaluation criteria are used as a starting point for the criteria selected for this benchmark.

The following sections detail the evaluation criteria recommended. The sections deal first with the general measurements followed by data communication specific measurements.

4.3.3.1 Benchmark General Evaluation Criteria

The overall system performance is important. The effect of data communication on power consumption and the effect of mobility on data communication are primary concerns. This data communication benchmark measures both. For power consumption, the average power consumed by clients and the average power consumed by servers is calculated.

The power consumption measure does not depend on the data communication method used. It requires the time each node spends in each CPU mode – transmit, receive, and standby during a simulation. The amount of power per unit time is calculated and averaged for all nodes of each type (LMH and SMH). The benchmark parameters are: *period, numSMH, numLMH, transRateS, recRateS, stbyRateS, transRateL, recRateL,* and *stbyRateL.* The evaluation parameters are *avgPwrS* and *avgPwrL*. These are the average power consumed by a SMH per unit time and the average power consumed by a LMH per unit time, respectively.

The protocol specific parameters are:

$trans_k$ – Time each node k spends in transmit mode.
rec_k – Time each node k spends in receive mode.
$stby_k$ – Time each node k spends in standby mode.

$$avgPwrS = \frac{\left(\sum_{k=1}^{k=numSMH} \left(\frac{(trans_k \times transRateS) + (rec_k \times recRateS) + (stby_k \times stbyRateS)}{period}\right)\right)}{numSMH} \tag{4-6}$$

$$avgPwrL = \frac{\left(\sum_{k=1}^{k=numLMH} \left(\frac{(trans_k \times transRateL) + (rec_k \times recRateL) + (stby_k \times stbyRateL)}{period}\right)\right)}{numLMH} \tag{4-7}$$

Power consumption is based on a unit time period. For each client and each server, the power consumed per time unit is calculated by multiplying the percentage of time in each mode by the cost in power dissipation of each power mode. This is done for each power mode – transmit, receive and standby. These are then summed and then divided by the time period the network ran. This gives us the cost per unit time for all clients and all servers. We calculate the average power consumed by dividing the totals by the number of nodes (SMH and LMH).

The effect of mobility that we measure is the percentage of SMHs out of range of all data broadcasts transmissions. This demonstrates the affect of network mobility and implies the level of node disconnection in the network. This criterion indirectly measures the percentage of SMHs able to benefit from any form of MANET data communication.

The benchmark parameter is *numSMH.* The evaluation criterion is *perCvr.* This is the average percentage of SMHs within range of a LMH's broadcast transmission.

The protocol variables are:

$numHeard_b$ – The number of SMHs detecting a LMH during synchronization in service cycle b.
numBcast – The number of broadcasts made during the simulation period.

$$perCvr = \frac{\left(\sum_{b=1}^{b=numBcast} \left(\frac{numHeard_b}{numSMH} \right) \right)}{numBcast} \tag{4-8}$$

Like power consumption, this measure can show a pattern of good or poor data service throughout the network. Next we consider measurements associated with specific data methods.

4.3.3.2 Benchmark Communication Type Specific Evaluation Criteria

MANET data communication includes data broadcast, data query and peer-to-peer communication. Criteria for each are discussed.

The broadcast portion of the MANET is important, as data push is energy efficient. The proposed measure to monitor this portion of data communication will be broadcast effectiveness. Broadcast effectiveness will be measured as an average for each broadcast and as an average for the entire simulation. As a client listens to the index at the beginning of each broadcast, it will count the number of items broadcast that are of interest to that client.

The benchmark parameters are $itemsBcast_k$ and *numSMH, where* k is the number of items broadcast to SMH k. The evaluation criteria are $bcastEff_m$ and *avgBcastEff*. The first measure, $bcastEff_m$, measures the effectiveness of broadcast m. The second measure, *avgBcastEff*, is the average effectiveness of all broadcasts during the simulation period.

The protocol specific variable for $bcastEff_m$ is:

$intItems_k$ – Number of items of interest to SMH k in broadcast.

$$bcastEff_m = \frac{\left(\sum_{k=1}^{k=numSMH} \left(\frac{\text{int}\, Items_k}{itemsBcast_k} \right) \right)}{numSMH} \tag{4-9}$$

The protocol specific variable for *avgBcastEff* is:

numBcast – The total number of broadcasts that occurred.

$$avgBcastEff = \frac{\left(\sum_{m=1}^{m=numBcast} \left(bcastEff_m \right) \right)}{numBcast} \tag{4-10}$$

The broadcast effectiveness is kept sufficiently general that it works when all broadcasts in a MANET are of the same length and also when each server has broadcasts of differing lengths. Requests not handled by the broadcast may need to be handled by the more expensive (energy-wise) data pull or peer-to-peer communication. A high broadcast effectiveness is desired. It means that client's needs are being met and that servers are not broadcasting unnecessary information.

The data pull section will rely on the measurement of query efficiency. This is a measure of the percentage of data queries that get served during a single service cycle or an average over an entire simulation.

The benchmark parameters are *reqFreq* and *numSMH.* The evaluation parameter is *queryEfficiency,* an overall system average.

The protocol specific parameter is:

totalServed$_k$ – *The total number of data queries sent by SMH k that were served.*

$$queryEfficiency = \frac{\left(\sum_{k=1}^{k=numSMH}\left(\frac{totalServed_k}{reqFreq}\right)\right)}{numSMH} \tag{4-11}$$

This measurement can be affected by the amount of disconnection in the network by lowering the number of queries served. The load on each LMH can also affect this measure if the load is more than the LMH can serve in the time allotted.

Peer-to-peer communication is a time when clients can communicate directly with clients and servers can communicate with servers. During this time, servers can perform coordination and control activities with other servers. In addition, clients can communicate

Peer efficiency can be measured by comparing the number of messages sent to peers by the number of messages received by peers. This is a system wide measurement. The benchmark parameters are *peerFreq* and *numSMH.* The evaluation criterion is *peerEfficiency.* This is a system wide average.

The protocol specific parameter is:

$msgRec_k$ – *The number of peer messages received by SMH k.*

$$peerEfficiency = \frac{\left(\sum_{k=1}^{k=numSMH} \left(\frac{msg\,Rec_k}{peerFreq} \right) \right)}{numSMH} \tag{4-12}$$

Equations 4-6 to 4-12 details the measurement made by the MANET data communication benchmark to compare and evaluate different MANET data communication protocols.

4.4 MANET Data Communication Benchmark Summary

In this Chapter a review of the background and purposes of benchmarks is presented. The specific parts of the MANET data communication benchmark are given. These are a standard architecture, workload, and set of evaluation criteria.

Before the benchmark was presented, each of these issues was examined as it related to current MANET Data Communication and Routing protocols. The MANET data communication benchmark was then presented, in three parts. First the benchmark architecture was presented followed by the benchmark workload. This was followed by the benchmark evaluation.

Without a standard benchmark, a comparison between proposed methods is difficult or impossible. The proposed benchmark provides a standard by which MANET data communication protocols might be measured and compared. As a benchmark must ultimately be developed and accepted by an entire community, this proposed benchmark is only a starting place for discussion. Over time, it is expected that this benchmark will be modified and altered until it gains general acceptance within the MANET data communication research community.

Chapter 5

EVALUATION OF TriM PROTOCOL

5.1 Introduction to Evaluation of TriM

This chapter provides an evaluation of TriM. The objective of this chapter is to evaluate the performance of the proposed MANET data communication protocol. We perform this evaluation in two ways. First, we evaluate the manner in which TriM addresses the MANET data communication issues raised in Chapter 1. This portion of the evaluation is contained in Section 5.2. Second, we evaluate the expected performance of TriM with respect to the benchmark developed in Chapter 4. Beginning with the benchmark evaluation equations given in Chapter 4, we use the protocol to define the anticipated behavior of TriM. This portion of the evaluation is found in Section 5.3. Section 5.4 summarizes the evaluation.

5.2 MANET Data Communication Issues Addressed

Section 1.5 presented the MANET data communication issues by first discussing the issues related to the MANET environment. Following the MANET data communication environmental issues, the issues related to each of the three methods of MANET data communication were addressed. In this section, TriM will be evaluated in the same order that the MANET data communication issues were discussed in Section 1.5.

In Section 1.5.3 a third type of MANET data communication issue was raised. This issue was the absence of a standard architecture, workload and set of evaluation criteria in current MANET data communication research. This third issue is addressed by the creation of a MANET data communication benchmark.

5.2.1 MANET Data Communication Environmental Issues

The first environmental issue is power consumption. This is an important issue as all nodes (LMHs and SMHs) are battery powered. Power consumption is considered by the proposed protocol in several ways.

First, the most efficient data communication method, data broadcast, is scheduled before other methods of data communication and is designed to eliminate contention for the wireless channel eliminating transmission collisions and the need for expensive broadcast retransmission.

Second, local control of all decision making eliminates an energy expensive and time consuming calculation of a leader. The leader of [37] was problematic as there was no expectation that LMHs within a leader's area could reach any or all of the SMHs. The Service Cycle will initially start with the deployment of all nodes, beginning with the first synchronization stage. The minimum amount of information needed for local decision-making is transmitted during synchronization. Each LMH and SMH tracks its own position within the Service Cycle time frame.

Third, the cost of calculating the popularity factor and Resident Latency of [37] is eliminated. The popularity factor is the number of nodes that have requested a data item in the past service cycle. Resident latency is not an issue as historical request data is not kept and data requests naturally expire. If a SMH has a data query not answered before the end of the Data Pull Stage and this data is also not broadcast during the following Data Push broadcast, the SMH will need to re-request this data.

This will only occur if the number of outstanding data requests during data pull exceeds the maximum size of the dynamic portion of the broadcast. If this occurs, data items will be selected in the order the queries were received.

Fourth, a single index is broadcast by each LMH as part of the data broadcast transmission. While this may have the effect of increasing data access time, the savings are in reducing the power consumed in data broadcast by reducing the amount transmitted.

Fifth, TriM puts each individual node in the lowest power consuming mode possible. Nodes only switch to transmit mode when necessary to transmit. When a node will not be active due to disconnection from the network, detected in the synchronization stage, the node switches to the least expensive mode of standby. At other times in the protocol, nodes switch to standby when they are idle.

Finally, neighbor node information is maintained by TriM for a single service cycle. This information is kept individually at each LMH/SMH and requires no overall coordination. This prevents the cost of tree-based algorithms, like Wieselthier [71].

The next issue has to do with node mobility. Node mobility can change the network topology and cause disconnections in the network. However, location information will remain useful for short periods of time. Maintaining location information for a single service cycle deals with the mobility of nodes by requiring the network to regularly update location information. The scenarios presented in Chapter 4 indicate that while nodes are mobile, nodes travel relatively slowly, less than 20 m/s. These movements can be ignored and the nodes treated as stationary for short periods of time. How short a period of time varies by scenario and is one factor that the network designer must consider when setting the network parameters during deployment.

While treated as stationary, there will be times when a SMH or LMH will move out of range, or otherwise be disconnected from the network. This could happen, for instance, as the power available to a node decreases. The end result of this assumption is that nodes detected during the synchronization stage will generally remain within range for an entire Service Cycle but this cannot be guaranteed. This is the primary reason that data queries and peer messages are not resent during a single service cycle. If a reply to a peer message or data query is not received one of two possibilities is most likely. First, the queried LMH or the SMH messaged may have moved out of range. Second, the queried LMH or SMH messaged may be overloaded and unable to get to the data query or peer message. In either case, a retransmission of the data query or peer message only serves to waste power.

The next environmental issue is timing. Two measures of timing are considered: access time and tuning time. Access time is a measure of the response time from data query to data service. Tuning time is a measure of the amount of time a node spends in transmit mode, the mode of greatest power consumption. Access time is addressed through having the dynamic portion of the broadcast. Items not handled during the Data Pull Stage will normally appear in the next broadcast. The result is that most or all data queries are handles within a single data pull stage and the any remaining data queries will be added to the next broadcast. The expectation is that under normal circumstances, all data queries are handled within one service cycle. If a network is overloaded, it is possible that the dynamic portion of the broadcast will be too small to transmit all of the pending data requests. When this occurs, the broadcast will include the maximum number of data items possible. However, in these cases there is no guarantee that every item will be broadcast within one Service Cycle. However, some degradation of performance is to be expected in overloaded networks. Minimum access time issue cannot be reasonably and fully provided in a MANET data communication protocol. The limited battery power of LMHs prevents immediate response to data queries as is possible in traditional wireless networks. Minimizing tuning time is a product of reducing the amount of time spent in transmit mode. This minimization was discussed earlier in this section.

The data integrity issue discussed in Section 1.5 was concerned with the database of each LMH having the same data values. Inconsistency in data values would mean that data provided to clients would vary depending on which LMH was queried. The data integrity issue is not addressed in this protocol, as a fully replicated database among all LMHs is a starting assumption. This assumption implies that all LMHs have identical databases throughout the simulation. The effect of a non-replicated database is an appropriate topic for later research.

5.2.2 MANET Data Communication Method Issues

There are issues involved in data broadcast, data query and peer-to-peer communication. Each will be discussed in turn.

During data broadcast, a variety of issues must be addressed. These can be summarized as broadcast content, SMH data needs, and index use. The broadcast content in this protocol is split between a static and dynamic portion. While it is recommended that the broadcast be kept as small as possible, no specific size is enforced. However, the broadcast is designed to include those data items specifically requested by SMHs in the previous service cycle. The static items are selected to be items needed by all nodes while the dynamic portion of the broadcast serves recent and irregular node data needs. A single index is used by a LMH for each broadcast. As discussed earlier, a single index can increase access time, but does so with extra power being consumed. The index is needed to allow SMHs to tune in and out of broadcasts for information that is needed while ignoring multiple transmission of the same data during a single service cycle. This allows SMHs to be in standby as much as possible during data broadcast. The index is a power saving feature SMHs. Too frequent transmission of the index wastes power for LMHs. The design of this protocol indicates a preference for reduced power consumption over reduced access time.

What to transmit during data broadcast is decided locally by each LMH. This can cause some data to be re-transmitted in the same service cycle. However, there is no guarantee that all SMHs can hear one LMH and not the other. Both need to broadcast the static portion of the broadcast. Each LMH will build the dynamic portion strictly from data items requested and not served during the most recent data query period. As SMHs query specific LMHs, the closest during the last synchronization, the size and content of the dynamic portion of the data broadcast will vary from LMH to LMH. The protocol also addresses the issue of how multiple requests for the same data item received by a single LMH are handled during the Data Pull Stage. Multiple requests for a single data item may be made by different SMHs. All duplicate requests received prior to the transmission of that data item are considered a single data request. A request for a data item after it has been transmitted is considered a new data request. This policy will eliminate the transmission of duplicative data queries.

The primary peer-to-peer issue addressed by TriM is the manner in which peer-to-peer communication is handled. Peer-to-peer messages not handled during a service cycle are dropped. Peer messages are not a data item served by the LMH. Routing of peer messages is a service provided to SMHs. Each time the network synchronizes, the topology of the network changes. It is possible that messages that previous required routing could be sent directly peer-to-peer.

TriM addresses the shortcomings of previous algorithms, addresses the general concerns of MANET database communication and permits all valid modes of MANET communication.

5.3 MANET Data Communication Benchmark Evaluation

When considering the anticipated performance of TriM, the benchmark proposed in Chapter 4 is used as our basis. The benchmark developed in Chapter 4 has five criteria for evaluation. These five criteria are average power consumption of LMHs and SMHs, percent of broadcast coverage, broadcast effectiveness, query efficiency and peer efficiency. In this section, these criteria for evaluation will be calculated for the proposed protocol of Chapter 3. These calculations will form the expected performance of the proposed MANET data communication protocol. Before calculating the benchmark parameters for each of our scenarios (military, rescue, and business) some additional information is needed. For all discussion here, we assume that the maximum number of LMHs and SMHs specified in the benchmark are used. The values used are from Tables 4-4, 4-5 and 4-6. The benchmark evaluation equations are in Equations 4-6 to 4-12.

Before discussing broadcast effectiveness we calculate the maximum percentage of the MANET region that can be reached by LMH data broadcasts. We assume no overlap of LMHs regions to calculate the maximum area of coverage. To calculate the maximum area of coverage, we calculate the area reached by a single LMH transmission and multiply by the number of LMHs. We then divide this number by the size of the roaming region, giving the maximum percentage of area covered by LMH broadcast transmission. Table 5-1 shows this maximum percentage of area covered. The most telling thing from this table is that if nodes are distributed throughout the region, the vast majority of the SMHs are out of range during data broadcast. This should lead to reduced power consumption for SMHs, as SMHs that do not detect a LMH during synchronization go into standby mode.

Scenario	# LMHs	Region Size	Percent Coverage
Military	20	10 km x 15 km	2.6%
Rescue	10	5 km x 5 km	7.85%
Business	6	1 km x 1 km	118%

Table 5-1 – Maximum Area of Data Broadcast Coverage

The other thing we notice from Table 5-1 is that LMHs in the business scenario potentially have full data broadcast coverage of the MANET region. As the maximum percentage of coverage is greater than 100%, it is possible that all or most SMHs can hear a data broadcast transmission. Power consumption among SMHs should increase as all or most of the SMHs will be in receive mode for at least one data broadcast. Few and possibly no SMHs will be in standby during the entire data push stage. In either case, LMH power is not affected as all LMHs transmit a broadcast every service cycle.

5.3.1 MANET Data Communication Evaluation – Power Consumption

The first benchmark evaluation criteria we consider are *avgPwrS* and *avgPwrL*. These benchmark criteria give an indication of the average power consumption of SMHs and LMHs respectively during the simulation. The power consumption is an average for all SMHs and all LMHs and is measured in units per time unit. For the protocol under consideration this is watts per hour, the units in which power dissipation rates are provided. The equations for *avgPwrS* and *avgPwrL* are given in Eqn 3-6 and 3-7 and are repeated in Sections 5.3.1.1 and 5.3.1.2 respectively. The equations use the amount of time spent in transmit, receive and standby modes during the simulation, the power dissipation rates in each mode, and the total time of the simulation. The parameters that vary from one simulation to the next are the amount of time spent in each power mode. We now look at the power consumption in each stage. If a node spends the majority of its time in one of the power nodes, the average power consumption should be close to that value.

5.3.1.1 SMH Average Power Consumption

The first benchmark evaluation criterion we consider is *AvgPwrS*. Equation 4-6 is:

$$avgPwrS = \frac{\left(\sum_{k=1}^{k=numSMH} \left(\frac{(trans_k \times transRateS) + (rec_k \times recRateS) + (stby_k \times stbyRateS)}{period} \right) \right)}{numSMH}$$

We simplify the equation for predicting performance of the protocol by calculating the average time spent in transmit, receive and standby for all SMHs during a service cycle. We cannot calculate the actual value for each SMH individually, so we calculate the average value for all SMHs. The equation then becomes Equation 5-1.

$$avgPwrS = \frac{(avgTrans \times transRateS) + (avg\mathrm{Re}c \times recRateS) + (avgStby \times stbyRateS)}{scLen} \quad (5\text{-}1)$$

To calculate the value for *avgPwrS,* the expected average power consumption per SMH, requires that we calculate *avgTrans, avgRec* and *avgStby*. The *scLen* is the length of one service cycle and varies by scenario and workload. All other values are provided as benchmark parameters. Once we calculate these values, we can determine the average expected power consumption of SMHs in the proposed protocol.

Starting with *avgTrans* we consider when a SMH will be in transmit mode. There are four stages to the proposed protocol. These are: synchronization, data push, data pull and idle. A SMH is never in transmit mode during the data push or idle stages. During synchronization, a SMH is in transmit mode long enough to transmit its synchronization data. The remainder of the synchronization mode is spent in receive mode. If a SMH hears no LMHs and no SMHs during synchronization, a SMH will be in standby mode during the entire data pull stage. If only SMHs are heard, a SMH may be in transmit mode when sending peer messages that need no routing, spending the remainder of the time in receive mode. If only LMHs are heard during synchronization, a SMH may transmit peer messages for routing and may transmit data queries, spending the remainder of the time in receive mode. If both LMHs and SMHs are heard, a SMH is in transmit mode during data pull when transmitting a data query or sending a peer messages, spending the remainder of the time in receive mode. Equation 5-2 gives the average amount of time each SMH k spends in transmit mode.

$$avgTrans = synchPart1 + pullPart1 \tag{5-2}$$

$$synchPart1 = transmit_{SMH}$$

$$pullPart1 = (probLS \times pullQP) + (probL \times pullQP) + (probS \times pullP)$$

$$pullP = (peerFreq \times transmit_{msg} \times pullLen)$$

$$pullQP = (reqFreq \times transmit_{qry} \times pullLen) + (peerFreq \times transmit_{msg} \times pullLen)$$

where:

avgTrans is the average time SMHs spend in transmit mode
peerFreq is the peer message frequency
probLS is the probability of a SMH hearing both LMHs and SMHs during synchronization
probL is the probability of a SMH hearing only LMHs during synchronization
probS is the probability of a SMH hearing only SMHs during synchronization
pullLen is the length of the pull stage
pullPart1 is the portion of avgTrans from the data pull stage
pullP is the transmit time for a SMH doing only peer messaging
pullQP is the transmit time for a SMH doing both data query and peer messaging
reqFreq is the data query frequency
synchPart1 is the portion of avgTrans from the synchronization stage
$transmit_{msg}$ is the time for a SMH to transmit one peer message
$transmit_{qry}$ is the time for a SMH to transmit one data query
$transmit_{SMH}$ is the time for a SMH to transmit its synchronization data.

We now consider the time spent in receive mode. No SMH is in receive mode during the idle stage. During synchronization, a SMH is in receive mode at all times except when transmitting its own synchronization data.

During data push, a node is in standby during the *bcastPrep* portion. If no LMHs were heard during synchronization, a SMH will be in standby mode during *bcastLen*, the time during which LMHs may transmit their data broadcasts in turn, as well. Otherwise, during *bcastLen* a SMH is in receive mode whenever listening to transmitted indices and items of interest. The SMH will listen to the index of every LMH heard during synchronization. Items of interest are the static portion of one LMH broadcast and zero or more items from the dynamic portion of broadcasts.

During Data pull, a SMH is in standby if no LMS and no SMHs were heard during synchronization. Otherwise, a SMH is in receive mode when not transmitting. The time spent in transmit mode is shown as *pullPart1* in Equation 5-2.

Equation 5-3 shows the calculation for the average time each SMH spends in receive mode.

$$avg\mathrm{Rec} = synchPart2 + pushPart2 + pullPart2 \tag{5-3}$$

$$synchPart2 = synchLen - transmit_{SMH}$$

$$pushPart2 = probB \times (indexTrans + staticTrans + \mathrm{int}Trans)$$

$$indexTrans = (items_{stat} + items_{dyn}) \times transmit_{idx} \times numHeard$$

$$staticTrans = items_{stat} \times transmit_{data}$$

$$\mathrm{int}Trans = probInt \times (items_{dyn} \times transmit_{data} \times numHeard)$$

$$pullPart2 = (probLS + probL + probS) \times (pullLen - pullPart1)$$

where:

avgRec is the average amount of time SMHs spend in receive mode
indexTrans is the time to transmit the index for each LMH
$items_{dyn}$ are the number of dynamic items in a broadcast
$items_{stat}$ are the number of static items in a broadcast
intTrans is the time to transmit items of interest
numHeard is the average number of LMHs heard by a SMH during synchronization
staticTrans is the time to transmit one static portion of a broadcast.
probB is the probability that a SMH will hear a LMH data broadcast
probInt is the probability that a dynamic data item will be of interest
probLS is the probability of hearing both LMHs and SMHs during synchronization
probL is the probability of hearing only LMHs during synchronization
probS is the probability of hearing only SMHs during synchronization
pullLen is the length of the data pull stage
pullPart1 is the portion of avgTrans from the data pull stage
pullPart2 is the portion of avgRec from the data pull stage
pushPart2 is the portion of avgRec from the data push stage
synchLen is the length of the synchronization stage
synchPart2 is the portion of avgRec from the synchronization stage
$transmit_{data}$ is the time for a LMH to transmit one data item
$transmit_{idx}$ is the time for a LMH to transmit one index item
$transmit_{SMH}$ is the time for a SMH to transmit its synchronization data

The last item to calculate for *avgPS* is *avgStby.* A SMH is never in standby during the synchronization stage. A SMH spends the entire idle stage in standby mode. This leaves the data push and data pull stages.

During data push, a SMH is in standby during the *bcastPrep*. During *bcastLen* a

SMH is in standby whenever it is not listening to an index or data transmission. A SMH is also in standby if no LMHs were detected during the synchronization stage. If we subtract the average amount of time spent in receive mode from the push stage length, we will have the average amount of time spent in standby during the data push stage. The time spent in receive mode during data push is calculated as *pushPart2* in Equation 5-3.

During data pull, a SMH is in standby if no LMHs and no SMHs were heard during data synchronization. The average amount of time the SMHs spend in transmit mode and in receive mode are calculated as *pullPart1* and *pullPart2* respectively. These average times are subtracted from the length of the data pull stage giving us the average amount of time spent in standby during data pull.

Equation 5-4 shows the calculation of *avgStby*.

$$\begin{gathered} avgStby = pushPart3 + pullPart3 + idleLen \\ pushPart3 = pushLen - pushPart2 \\ pullPart3 = pullLen - (pullPart1 + pullPart2) \end{gathered} \tag{5-4}$$

where:

avgStby is the average amount of time SMHs spend in standby mode
idleLen is the length of the idle stage
pullLen is the length of the data pull stage
pullPart1 is the portion of avgTrans from the data pull stage
pullPart2 is the portion of avgRec from the data pull stage
pullPart3 is the portion of avgStby from the data pull stage
pushLen is the length of the data push stage
pushPart2 is the portion of avgRec from the data push stage
pushPart3 is the portion of avgStby from the data push stage

By replacing *avgTrans, avgRec* and *avgStby* from Equations 5-2 to 5-4 into Equation 5-1, we calculate the expected power consumption of a SMH per unit time for the protocol proposed. In Chapter 7, Equation 5-1 is calculated making probability estimates based on the simulation assumptions and is compared to the actual simulation results.

5.3.1.2 LMH Average Power Consumption

The next benchmark evaluation criterion we consider is *AvgPwrL*. Equation 4-7 is:

$$avgPwrL = \frac{\left(\sum_{k=1}^{k=numLMH} \left(\frac{(trans_k \times transRateL) + (rec_k \times recRateL) + (stby_k \times stbyRateL)}{period} \right) \right)}{numLMH}$$

We again simplify the equation for predicting performance of the protocol by calculating the average time spent in transmit, receive and standby for all LMHs. We cannot calculate the actual value for each LMH individually, so we calculate the average value for all LMHs. The equation then becomes Equation 5-5.

$$avgPwrL = \frac{(avgTrans \times transRateL) + (avg\mathrm{Re}c \times recRateL) + (avgStby \times stbyRateL)}{period} \tag{5-5}$$

To calculate the value for *avgPwrL* requires that we calculate *avgTrans, avgRec* and *avgStby* for LMHs. We do this in the same manner as we did for SMHs. Once we calculate these values, we can determine the average expected power consumption of LMHs in the proposed protocol.

Starting with *avgTrans* we consider when a LMH will be in transmit mode. There are four stages to the proposed protocol. These are: synchronization, data push, data pull and idle. A LMH is never in transmit mode during the idle stage. During synchronization, a LMH is in transmit mode long enough to transmit its synchronization data. The remainder of the synchronization mode is spent in receive mode.

During data push, a LMH is in transmit mode while transmitting its data broadcast, with associated index. The remainder of data push is spent in either receive or standby mode. The remaining stage is data pull. If a LMH heard no LMHs and no SMHs during synchronization, the LMH will be in standby. If only LMHs were heard during synchronization, a LMH must be available for peer message routing. If only SMHs are heard during synchronization, a LMH must be available for data query. Otherwise the LMH will be in receive mode and ready to route peer message and handle data queries. A LMH in receive mode will switch to transmit mode to transmit data query responses and to route peer messages.

The calculation for *avgTrans* is shown in Equation 5-6.

$$avgTrans = synchPart1 + pushPar1 + pullPart1 \tag{5-6}$$

$$synchPart1 = transmit_{LMH}$$

$$pushPart1 = (items_{stat} + items_{dyn}) \times (transmit_{idx} + transmit_{data})$$

$$pullPart1 = (probLS \times pullQP) + (probL \times pullP) + (probS \times pullQ)$$
$$pullP = (peer\mathrm{Re}c \times transmit_{rte} \times pullLen)$$
$$pullQ = (qry\mathrm{Re}c \times transmit_{data} \times pullLen)$$
$$pullQP = (qry\mathrm{Re}c \times transmit_{data} \times pullLen) + (peer\mathrm{Re}c \times transmit_{rte} \times pullLen)$$

where:

avgTrans is the average time LMHs spend in transmit mode
$items_{dyn}$ is the number of dynamic items in a LMH broadcast
$itmes_{stat}$ is the number of static items in a LMH data broadcast
peerRec is the number of routing requests received at a LMH
probLS is the probability of a SMH hearing both LMHs and SMHs during synchronization
probL is the probability of a SMH hearing only LMHs during synchronization
probS is the probability of a SMH hearing only SMHs during synchronization
pullLen is the length of the pull stage
pullPart1 is the portion of avgTrans from the data pull stage
pushPart1 is the portion of avgTrans from the data push stage
pullP is the transmit time for a SMH doing only peer messaging
pullQ is the transmit time for a SMH doing both data query and peer messaging
pullQP is the transmit time for a SMH doing both data query and peer messaging
qryRec is the number of data queries received at a LMH

synchPart1 is the portion of avgTrans from the synchronization stage
$transmit_{data}$ is the time for a SMH to transmit one peer message
$transmit_{idx}$ is the time for a SMH to transmit one data query
$transmit_{LMH}$ is the time for a SMH to transmit one data query
$transmit_{rte}$ is the time for a SMH to transmit its synchronization data.

We now consider the time spent in receive mode. No LMH is in receive mode during the idle stage. During synchronization, a LMH is in receive mode at all times except when transmitting its own synchronization data.

During data push, a LMH is in receive mode during the *bcastPrep* portion. During *bcastLen*, a LMH is either transmitting or in standby mode. During data pull, a LMH is in standby if no LMS and no SMHs were heard during synchronization. Otherwise, a LMH is in receive mode when not transmitting. The time spent in transmit mode is shown as *pullPart1* in Equation 5-6.

Equation 5-7 shows the calculation for the average time each LMH spends in receive mode.

$$avg\mathrm{Re}c = synchPart2 + pushPart2 + pullPart2 \tag{5-7}$$
$$synchPart2 = synchLen - transmit_{LMH}$$
$$pushPart2 = bcast\,\mathrm{Pr}\,ep$$
$$pullPart2 = (probLS + probL + probS) \times (pullLen - pullPart1)$$

where:

avgRec is the average amount of time LMHs spend in receive mode
bcastPrep is the length of the broadcast preparation portion of data push
probLS is the probability of hearing both LMHs and SMHs during synchronization
probL is the probability of hearing only LMHs during synchronization
probS is the probability of hearing only SMHs during synchronization
pullLen is the length of the data pull stage
pullPart1 is the portion of avgTrans from the data pull stage
pullPart2 is the portion of avgRec from the data pull stage
pushPart2 is the portion of avgRec from the data push stage
synchLen is the length of the synchronization stage
synchPart2 is the portion of avgRec from the synchronization stage
$transmit_{LMH}$ is the time for a SMH to transmit its synchronization data

The last item to calculate for *avgPwrL* is *avgStby.* A LMH is never in standby during the synchronization stage. A LMH spends the entire idle stage in standby mode. This leaves the data push and data pull stages.

During data push, a SMH is in standby during the *bcastLen* when not transmitting its data broadcast and associated index. During data pull, a LMH is only in standby if no LMHS and no SMHs were heard during data synchronization.

Equation 5-8 shows the calculation of *avgStby.*

$$avgStby = pushPart3 + pullPart3 + idleLen \tag{5-8}$$

$$pushPart3 = bcastLen - pushPart1$$

$$pullPart3 = [1 - (probLS + probL + probS)] \times pullLen$$

where:

avgStby is the average amount of time SMHs spend in standby mode
bcastLen is the amount of time in the data push stage for transmission of broadcasts
bcastLen is the amount of time in the data push stage for the transmission of broadcasts
probLS is the probability of hearing both LMHs and SMHs during synchronization
probL is the probability of hearing only LMHs during synchronization
probS is the probability of hearing only SMHs during synchronization
pullLen is the length of the data pull stage
pullPart3 is the portion of avgStby from the data pull stage
pushPart1 is the portion of avgTrans from the data push stage
pushPart3 is the portion of avgStby from the data push stage
synchLen is the length of the synchronization stage

By replacing *avgTrans, avgRec* and *avgStby* from Equations 5-6 to 5-8 into Equation 5-5, we calculate the expected power consumption of a LMH per unit time for the proposed protocol. In Chapter 7, Equation 5-5 is calculated for each MANET scenario, making probability estimates based on the simulation assumptions and is compared to the actual simulation results.

It is clear that a number of factors affect the amount of power consumed by a LMH. However, unless the idle stage is very long when compared to the rest of the service cycle it would appear that a LMH that is active during data pull spends the majority of its time in receive mode and a smaller amount of time in transmit mode. An active LMH spends little time in standby mode. As a LMH that is inactive during data pull is in standby mode, an inactive LMH will consume less power.

Just like with a LMH there are a number of factors affect the amount of power consumed by a SMH. A SMH spends much more time on average in standby mode than a LMH. When no LMHs are detected during synchronization and no LMHs and no SMHs are detected during synchronization, a SMH can spend almost the entire service cycle in standby.

5.3.2 MANET Data Communication Evaluation – Network Connectivity

Network connectivity is measured as the percentage of SMHs that are in range of a LMH data broadcast transmission. The benchmark evaluation criterion is given in Equation 4-8, and is repeated here:

$$perCvr = \frac{\left(\sum_{b=1}^{b = numBcast} \left(\frac{numHeard_b}{numSMH} \right) \right)}{numBcast}$$

This equation averages the ratio of SMHs hearing a broadcast to the number of SMHs in a network over the life of the network. It is assumed that nodes are initialized in random locations, within a defined region, for all scenarios and that the nodes move randomly. If we assume random distribution of nodes, then the *perCvr* for any broadcast should be similar to the overall simulation average. This gives us:

$$perCvr = \frac{numHeard_b}{numSMH}$$

We further assume that the number of SMHs able to hear a data broadcast when

SMHs are distributed randomly throughout the regions can be approximated by the percentage of the region covered by LMH broadcast transmissions. If network connectivity is good, the percentage of SMHs able to hear a data broadcast transmission should be high. The maximum broadcast transmission coverage in each scenario is shown in Table 5-1. As can be seen in Table 5-1, this percentage is very low for the military scenario and for the domestic rescue scenario. Equation 5-9 gives the expected average number of SMHs able to hear a LMH data broadcast based on this discussion.

$$perCvr = \frac{transArea \times numLMH}{regionWidth \times regionHeight} \tag{5-9}$$

where:

perCvr is the maximum percentage of area covered by LMH broadcast transmission, and represents the percentage of SMHs hearing a broadcast transmission
transArea is the area one LMH transmission can cover, assuming 360° of transmission
numLMH is the number of LMHs
regionWidth and regionHeight are the dimensions of the network region

When the percentage of coverage is low, the percentage of those capable of hearing a data broadcast transmission is, on average, equally low for a set of randomly dispersed nodes.

5.3.3 MANET Data Communication Evaluation – Broadcast Effectiveness

Assuming that SMHs are distributed randomly in the military and rescue scenarios, A LMH will typically have a very small number of SMHs that hear a broadcast transmission and few if any SMHs will hear more than one LMH data broadcast. In this case, the broadcast effectiveness will be very high for the few

SMHs hearing the broadcasts as they will typically hear the static portion only one time per data push stage. In addition, with only one or a few SMHs competing for data service during the previous data push cycle, the number of items not served should be small, possibly zero. The items in the next data broadcast transmission will be those items the SMH within transmission range of the LMH. The benchmark equations for broadcast effectiveness is given in Equation 4-10 and is:

$$avgBcastEff = \frac{\left(\sum_{m=1}^{m=numBcast} \left(bcastEff_m \right) \right)}{numBcast}$$

The $bcastEff_m$ is given in equation 4-9 and is:

$$bcastEff_m = \frac{\left(\sum_{k=1}^{k=numSMH} \left(\frac{\text{int}\, Items_k}{itemsBcast_k} \right) \right)}{numSMH}$$

The amount of overlap between LMHs should have the effect of lowering the broadcast effectiveness of LMHs, as SMHs will only need one copy of the static portion of the broadcast even if several are transmitted. The question in this section is how to estimate the broadcast effectiveness based on these equations. Upon evaluation of Equation 4-9 we see that broadcast effectiveness in a single broadcast is the ratio of wanted data items heard to the total number of items heard. The average broadcast effectiveness is the average of these ratios.

Assuming random distribution of the LMHs and SMHs, the expected broadcast effectiveness should be similar from one broadcast to the next. Broadcast effectiveness can be estimated as shown in Equation 5-10:

$$avgBcastEff = \frac{(items_{stat} + [probInt \times (items_{dyn} \times numHeard])}{(items_{stat} + items_{dyn}) \times numHeard} \quad (5\text{-}10)$$

where:

avgBcastEff is the expected Broadcast Effectiveness
numHeard is the average number of LMHs heard during synchronization mode
$items_{dyn}$ are the number of dynamic items in a data broadcast
$items_{stat}$ are the number of static items in a data broadcast
probInt is the probability that a dynamic data item is of interest to a SMH

Assuming the percentages of broadcast coverage as shown in Table 5-1, there is little to no overlap in the battlefield and rescue scenarios. If this is correct, the broadcast effectiveness for these scenarios should be close to 100%, with the broadcast effectiveness of the business scenario lower than the other two. The broadcast effectiveness of the battlefield and rescue scenarios should be essentially the same.

The difference between the three scenarios is that the percentage of SMHs served by broadcast in a battlefield and rescue scenario will be small, while the majority of SMHs will be served by data broadcast in the business scenario.

5.3.4 MANET Data Communication Evaluation – Query Efficiency

In the previous section we discussed broadcast effectiveness. This measures one of the three methods of MANET data communication. The second method of data communication is data query. We indirectly measure the effectiveness of data query when we measure the connectedness of the network. We more directly measure the effectiveness of data query by calculating the percentage of data queries that are served in a single service cycle or the average percentage of data queries served over the length of a simulation. The equation for query efficiency is given in Equation 4-11, repeated here:

$$queryEfficiency = \frac{\left(\sum_{k=1}^{k=numSMH}\left(\frac{totalServed_k}{reqFreq}\right)\right)}{numSMH}$$

When evaluating the expected performance of the proposed protocol on query efficiency, we use the probability that a SMH is within transmission range of a LMH to estimate the number of queries served. The estimate for query efficiency is given in Equation 5-11.

$$queryEfficiency = \frac{probR \times reqFreq \times numSMH}{reqFreq \times numSMH} \qquad (5\text{-}11)$$

where:

numSMH is the number of SMHs in the system
probR is the probability that a SMH transmission can reach a LMH
reqFreq is the number of requests submitted by each SMH

The estimate for query efficiency reduces to *probR*, the probability that a LMH is close enough to hear a data query. As the length of a data pull stage increases with the average number of SMHs served by each LMH, a LMH will typically be able to serve all data queries received. The query efficiency then measures the likelihood that a query will be heard and processed.

The actual amount of time it takes a LMH to process a query is not given in the benchmark or protocol. This value is dependent on the processor speed and DBMS in use as well as other factor outside the control of the protocol. A data query could be minimally served in the time it takes to transmit a query, process the query and transmit the response. The time it takes a SMH to transmit the query is fixed. The time it takes a LMH to transmit the response is also fixed. These times are both the direct result of LMH/SMH bandwidth and packet size. What we are measuring is the number of queries that are received at a LMH and of those received, how many we can respond to during a single data pull stage.

If a query can be immediately processed, and there is time to transmit a response, we assume the query is served. If the number of SMHs submitting data queries to a single LMH increases beyond the expected threshold, the queries may get delayed, as insufficient time may exist in the data pull stage to transmit all responses. The protocol specifies that a SMH will query the LMH that is closest during the synchronization stage. This prevents SMHs from querying any random LMH. If LMHs and SMHs are distributed randomly and approximately uniformly, each LMH will serve approximately the same number of SMHs. However, this is not simply a matter of dividing the number of SMHs with the number of LMHs. As Table 5-1 indicates, the SMHs may easily be outside the reach of a LMH and is therefore incapable of submitting a data query. This does not negate the fact that these SMHs outside the reach of a LMH still have data queries to send.

If the actual load on the LMHs is too high, actual query efficiency will decrease as the number of queries not served rises. If a large percentage of SMHs are expected to be unable to transmit queries to a LMH, expected query efficiency will be low. If this expectation does occur, the actual query efficiency will be low as well.

When considering the effect of network disconnection on the percentage of data requests satisfied, it is clear that disconnection can have a significant effect. The protocol provides sufficient time for the average number of SMHs in reach of a LMH to transmit some set number of data queries. As long as the number of nodes remains close to this average, as it would if nodes are randomly distributed, the percentage of requests satisfied for the nodes within reach of a LMH should remain high. However, if a large percentage of SMHs are not within reach of a LMH or other

SMHs, the data requests of these SMHs will go unserved. Congestion of SMHs around a single LMH would also affect the percent of queries satisfied. However, the effect of periodic congestion in a random mobility model would be small compared to the effect of a high percentage of SMHs disconnected from the network.

5.3.5 MANET Data Communication Evaluation – Peer Efficiency

Peer efficiency will be affected by disconnection from the network in a similar fashion to query efficiency. Here, any queries that must be routed require the presence of a LMH. Queries sent directly only rely on the presence of the recipient SMH in the previous synchronization stage. The equation for peer efficiency is given in Equation 4-12, repeated here:

$$peerEfficiency = \frac{\left(\sum_{k=1}^{k=numSMH}\left(\frac{msg\operatorname{Rec}_k}{peerFreq}\right)\right)}{numSMH}$$

When estimating peer efficiency we must consider the times that no peer messages can be sent. If SMH only hears other SMHs during synchronization, those SMHs can still be sent a peer. Routed peer messages cannot be sent. If a SMH only hears LMHs, a peer message can be routed. Direct peer messages cannot be sent. If a SMH detects both LMHs and SMHs during synchronization peer messages can be sent directly and also can be routed. The only time peer messages cannot be sent is when no LMHs and no SMHs are detected during synchronization.

The calculated peer efficiency is given in Equation 5-12.

$$peerEfficiency = \frac{(probLS \times peerFreq) + (probL \times peerRte) + (probS \times peerMsg)}{peerFreq} \tag{5-12}$$

where:

peerRte is the average number of peer messages routed
peerMsg is the average number of peer messages sent directly
peerFreq is the average number of peer requests sent. peerFreq = peerRte+peerMsg
probLS is the probability of a SMH hearing both LMHs and SMHs during synchronization
probL is the probability of a SMH hearing only LMHs during synchronization
probS is the probability of a SMH hearing only SMHs during synchronization

As the coverage of LMHs in the military and rescue scenarios is small, an increase in the percentage of messages that need to be routed will decrease peer efficiency in these networks.

5.4 Summary of MANET Data Communication Protocol Evaluation

In this chapter, the proposed MANET data communication protocol has been evaluated. This evaluation has taken two forms. First, the MANET data communication issues discussed in Chapter 1 we addressed. A MANET data communication protocol must address many or all of these issues in order to be effective. The manner in which the proposed protocol addresses the issues was examined along with the rationale for the method in which the proposed protocol coordinates the three methods of MANET data communication.

Addressing these issues was an important first step in the evaluation of the proposed MANET data communication protocol. However, that alone is insufficient. Following the evaluation of how the proposed protocol addresses the general MANET data communication issues, the performance of the protocol was addressed. The MANET data communication benchmark proposed in Chapter 4 formed the basis of this evaluation. As part of that benchmark, five criteria were proposed for the evaluation of any MANET data communication protocol. The MANET data communication protocol proposed in this research was evaluated with respect to these benchmark parameters. The expected performance of this protocol with respect to the benchmark evaluation parameters is given. In Chapter 7, these projections will be compared to the results obtained from simulating the proposed MANET data communication protocol for each of the three scenarios presented (battlefield, rescue, and business). Conclusions and directions for further research will be drawn from this comparison.

Chapter 6

SIMULATION OF TriM PROTOCOL

6.1 Introduction

In Chapter 5, a more mathematical approach is used to describe the operation and behavior of the protocol. In this chapter, the protocol is simulated using the AweSim simulation software [62b]. This simulation will use the architecture and workload of the MANET data communication benchmark. The simulation will be used to determine simulated values for the benchmark evaluation parameters. Specifically, for each scenario we will be calculating average SMH and LMH power consumption, percent of SMHs not hearing a data broadcast, broadcast effectiveness, query efficiency and broadcast efficiency.

Simulation has been done in previous studies, such as [30][37], to study, evaluate and validate routing and data communication protocols. Simulation is a valuable tool when developing data communication protocols. In this research, the purpose of the simulation is to evaluate the protocol in terms of the proposed benchmark. The simulation allows the comparison of this protocol under a variety of conditions for each of the three scenarios: battlefield, rescue, and temporary business networks. The simulation will also provide the necessary data to compare this protocol to the work of Gruenwald [37] and Wieselthier [72].

In this chapter, the design of the simulation model will be presented. This discussion will deal first with the overall simulation model and assumptions made as part of the simulation. Following this, a detailed description of the simulation is given. Finally the results for the simulation are provided and analyzed.

6.2 Description of the Simulation Model

The AweSim simulation software [62b] using inserts coded in the C programming language to describe network behavior was used for this simulation study. AweSim is a general-purpose simulation tool that provides discrete event simulation of user defined networks [61b].

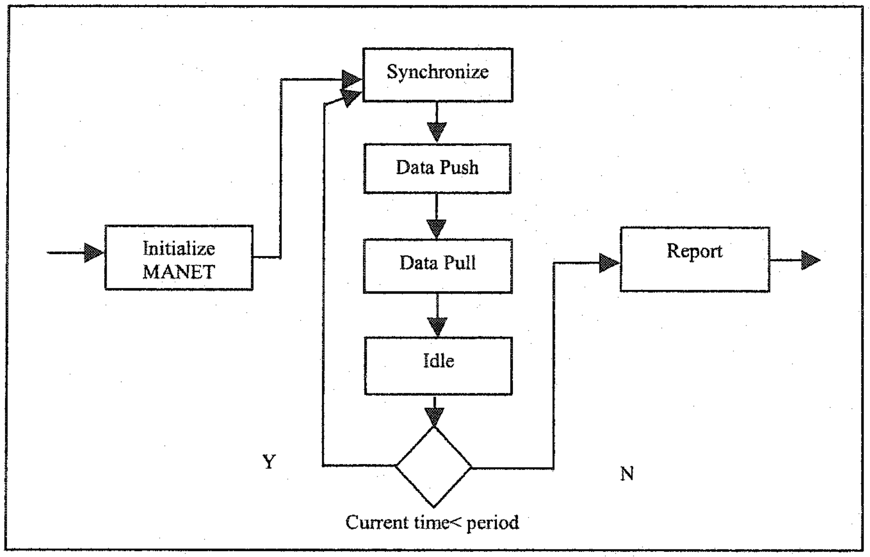


Figure 6-1 MANET Simulation Model

The model used to simulate the proposed MANET data communication protocol is shown in Figure 6-1. The deployment of a MANET and the execution of the proposed MANET data communication protocol can be defined as a set of discrete events that occur during the operation of the protocol. These events are network deployment and initialization, execution of the MANET data communication service cycle (SC), and a network report event that executes once at the conclusion of network simulation. The SC can further be modeled as a sequence of discrete events that repeat throughout the life of the network deployment, or in our case, the network simulation. These discrete events directly map to the stages of the proposed protocol namely synchronization, data push, data pull, and idle. For each event, code inserts in C are provided to describe the operation of the protocol. The code inserts are provided in Appendix B.

For each of the three scenarios (battlefield, rescue and business networks) data push parameters and data pull parameters are varied. Each of the three data broadcast sizes (50, 100 and 200 items) is simulated for each of the data pull parameters. These broadcast sizes are referred to in the results as small, medium and large broadcasts respectively. The data pull parameters are the frequency of *data query and peer messaging. Data query and peer messaging are* set to the same value. As both data query and peer message frequency are set to the same value, they are referred to collectively as pull frequency throughout this chapter. The values used for pull frequency are 5, 20 and 40 items/sec. These are referred to as low, medium and high pull frequency respectively. This variation simulates different loads on the network. These are discussed in Chapter 4 as part of the benchmark workload. This results in nine different workloads for each of the three scenarios. Each of the nine workloads were simulated 10 times. The same nine workloads were repeated for each of the three scenarios. An average of each of the 27 scenario/workload situations is shown in Sections 6.3.2 to 6.3.4. In addition to the benchmark scenarios, the number of LMHs and the transmission radius of nodes are varied in a set of simulations to show the behavior of the proposed protocol. These are described in Section 6.3.5.

Each of the discrete events that make up the simulation will be described. These are Section 6.2.2 – Network Deployment Event, Section 6.2.3 – Service Cycle Event and Section 6.2.4 – Recording Event.

6.2.1 Simulation Assumptions

As with any simulation, some assumptions must be made for certain parameters. These assumptions are given here, with their rationale. These assumptions are also discussed in the appropriate protocol stage.

In synchronization, five bytes were allocated for LMH synchronization data. The LMH data transmitted is the LMH ID and the x and y location of the node. The location is stored as an integer, indicating the number of meters from the origin. As the largest region is 15000 meters, 2 bytes are sufficient to store the x position and 2 bytes to store the y position. As the LMH indices range from 1 to 20, a single byte is sufficient to store that value. A SMH requires 6 bytes for synchronization data. The same 4 bytes are needed to store the x and y location of each node. However, there can be as many as 1000 nodes. Two bytes are necessary to store the unique ID for each SMH. The LMH then needs 5 bytes for synchronization data and the SMH needs 6 bytes.

During the SMH portion of synchronization, the average number of SMHs within reach of each LMH must be calculated. As the nodes are distributed randomly in the region, the average used is *numSMH / numLMH*.

The next stage is the data push stage. In this stage, a broadcast is built. An assumption must be made on the percentage of each broadcast transmission that is reserved for static data items and dynamic data items. In this simulation we assume that the broadcasts are equally split between static and dynamic data items. This means that a broadcast is always at least half full, even if no dynamic items need inclusion in the broadcast. It is further assumed that a SMH listens to the static portion of one broadcast transmission and to the entire dynamic portion of each broadcast transmission in its region.

In data pull we assume a static number of data queries and peer messages per node. The values used in the simulation are request frequencies of 5, 20 and 40 queries/message per second. If a node cannot transmit a query or send a peer message because no nodes were detected during synchronization, it is still assumed that the SMH desired to make those transmissions when calculating effectiveness of data query and peer messaging. During simulation, the distance between nodes is calculated and compared to benchmark transmission ranges to determine is a SMH can hear a LMH and if a SMH transmission can reach other nodes. It is assumed that a SMH will send all data queries and routing requests to the closest LMH detected during synchronization if within SMH transmission range.

For this initial simulation, the idle period is set to 0. The Gruenwald and Weiselthier protocols do not have an idle period. By setting this value to 0, we can more accurately compare these protocols to the protocol proposed in this research.

6.2.2 The Network Deployment Event

Prior to the network deployment event, a large number of values are defined for the protocol. These values are shown in Table 6-1. These values, while setting simulation values, may change from one scenario to the next to allow the simulation of a number of potential situations. When values that vary between scenarios all are listed in Table 6-1. The values in this table are used to calculate the amount of time for a LMH to transmit a data item, index item, data query or to route a peer-to-peer message. These calculations are based on the size of the item transmitted and the bandwidth of the LMH. Similarly, the time for a SMH to transmit a data query or a peer-to-peer message can be calculated based on the size of the item and the bandwidth of a SMH.

Parameter	Description	Default Value
numLMH	The number of LMHs	Military – 20 Rescue – 10 Business – 6
numSMH	The number of SMHs	Military – 1000 Rescue – 50 Business – 1000
transRateL recRateL stbyRateL	LMH Power Dissipation Rate in watts per hour Transmit / Receive / Standby	170 / 20 / 2
transRateS recRateS stbyRateS	SMH Power Dissipation Rate in watts per hour Transmit / Receive / Standby	7 / 1 / 0.1
BandL BandS	Bandwidth for LMH / SMH	2 Mbps 100 kbps
MinMob MaxMob	Minimum and maximum mobility for all nodes in meters per second	Military – 0 to 20 Rescue – 0 to 10 Business – 0 to 1
Period	Length of simulation in seconds	3600
RegionX RegionY	The size of the region in meters (X direction and Y direction)	Military – 10x15 km Rescue – 5x5 km Business – 1x1 km
IdxSize Data Size QrySize MsgSize	The size of one index item, one data item, one data query and one peer-to-peer message in bytes	128 64 256 512
SynchSizeL SynchSizeS	The size of the synchronization message for LMHs and SMHs in bytes	5 6
cpuL cpuS	CPU processing power in MIPS	1700 100

Table 6-1 – Primary Simulation Values

This event provides for the deployment of a MANET with a set of LMHs and SMHs initialized with an appropriate set of parameters. The parameters used to initialize the simulation protocol and guide its operation come from two locations in previous chapters. In Chapter 4, the proposed benchmark provides architecture and workload parameters. These are used in the initialization and operation of this simulation model. In addition, parameters associated with the proposed protocol are provided in Chapter 3. These parameters and the guidelines for recommended initial values are used in the simulation model. A very few additional parameters are used in the operation of the simulation. They will be discussed in the appropriate section. Parameters specific to network initialization are summarized in Table 6-2.

As previously mentioned, the network deployment event has two responsibilities. During this event the initial configuration for all LMHs and SMHs is set. In addition, all parameters needed to simulate the protocol are also set. Each LMH is initialized with a starting location (x and y coordinates) and the initial power level. In addition, the total time spent in each node power mode (transmit, receive, and standby), the number of broadcasts, and the number of dynamic items in the next broadcast are initialized to zero. Each node is also designated as active. Active indicates that a node can participate in the synchronization stage. All nodes having power are active at the start of the synchronization stage. The typical values used in the simulation are provided in Table 6-2. These values are calculated for each scenario/workload.

Parameter	Description	Typical Values
simTime	The amount of time that has elapsed in the simulation	initially 0
synchL synchS synchLen	The time needed to perform LMH synchronization, and SMH synchronization. synchLen = synchL + synchs	0.00305 sec
bcastPrep bcastLen pushLen	The time needed to prepare broadcasts and to transmit all index/broadcasts	32.06400 sec
pullLen	The time for data pull (data query and peer-to-peer messages)	8.00000 sec
idleLen	The time spent idle	0 sec.
FDBaseSz BcastSz	The size of the database and the maximum size of a broadcast	50 / 500 100 / 2000 200 / 5000
ReqFreq PeerFreq	The number of data requests and peer messages each SMH desires to transmit per second	5 / 20 / 40

Table 6-2 – Network Deployment Parameters for Simulation

The SMHs are initialized in a similar fashion. The initial location and power level are set. In addition, time spent in each power mode is set to zero. Just as the LMH tracks the number of broadcasts, the SMH track information about queries sent and responses to queries received as well as peer messages sent and received. These values are also initialized to zero. Each SMH is set as active.

Several variables controlling the simulation are also set. First, the simulation time is set to 0, indicating that no time has elapsed. For the synchronization stage, the LMH synchronization period, SMH synchronization period and total time for synchronization are calculated. For data push, the time necessary for broadcast preparation and index/broadcast transmission as well as the length of the data push stage are calculated. The length of the data pull stage is calculated and the length of the idle stage of the protocol is set.

The method for setting these values was discussed in Chapter 3. Briefly, the *synchL* is the amount of time for each LMH to transmit their ID and location in turn. Te *synchS* is the amount of time needed for the number of SMHs typically in the broadcast region of a LMH to transmit their ID and location. What this value is set at for this simulation is discussed under assumptions in Section 6.2.4. During data push, *bcastPrep* is the time to create an index and broadcast of maximum length. The *bcastLen* is the amount of time for each LMH to transmit an index and broadcast of maximum length, in turn. The size of the database and the maximum size of the broadcast are being varied as part of this simulation, as shown in Table 6-2. During data pull, *pullLen* is set to allow the average number of SMHs a LMH can hear to transmit a set number of data queries. The average number is discussed under assumptions in Section 6.2.4. The number of data queries is a parameter that is being varied as part of this simulation, as is the frequency of peer requests.

6.2.3 The Service Cycle Event

The service cycle (SC) repeats throughout the simulation. Within the SC are four discrete events that correlate to the four stages of the data communication protocol. The parameters used in these four stages can be found in Tables 6-3 to 6-6, in the following sub-sections. As with network deployment, these parameters are primarily provided by the benchmark of Chapter 4 and the protocol description of Chapter 3. The simulation repeats, gathering the data necessary to make the benchmark evaluation calculations until the simulation time expires. The simulation begins at time 0. As each stage of the protocol completes, the amount of time taken by that stage is added to simulation time. After the idle stage in each service cycle, the simulation time is compared to *period*. If the simulation time is greater than *period*, set in the benchmark as 3600 seconds, the simulation exits the service cycle and proceeds to the reporting event prior to termination.

6.2.3.1 The Synchronization Stage Event

During the synchronization stage we calculate the values that would be determined at each node during actual synchronization. Specifically, the ID and location of each node near enough to be heard are calculated and stored. This is done by calculating the distance between each pair of nodes, comparing the calculated distance with the transmission ranges of LMHs and SMHs. The simulation calculates the nodes that are heard and the nodes that are reachable. For instance, a SMH 200 meters from a LMH can hear that LMH because the transmission range of a LMH is 250 meters. However, the same SMH cannot reach the LMH because the transmission range of a SMH is 100 meters. In this situation a SMH can hear the LMH's broadcast transmission but cannot send data queries to that LMH. This distinction becomes important during data query, as there is no reason to transmit a query to a LMH too far away to hear the transmission.

SMHs that detect no LMH become inactive during synchronization, entering standby mode. This means that the calculated distance between the SMH and all LMHs is greater than the transmission range of LMHs.

The amount of time each LMH spends in transmit and receive mode is calculated and stored. The amount of time each SMH spends in transmit and receive node is also calculated and stored. These values are calculated using the size of the synchronization data and the bandwidth of either the LMH or SMH. The time spent in transmit, receive and standby mode is maintained for each LMH and SMH. The time each node spends transmitting their synchronization data is added to time spent in transmit mode. The remainder of the synchronization time is added to receive mode for every node. The amount of simulation time increases by *synchLen.*

6.2.3.2 The Data Push Stage Event

During data push every LMH spends the *bcastPrep* time in receive mode, building the index and data broadcast. During the *bcastLen*, each LMH is in standby, except when transmitting. The *bcastPrep* time is added to the receive time for every LMH and standby time for each SMH.

Each active SMH is in standby except during the transmission time for each LMH detected during synchronization. A SMH will be in receive mode for the index of each

LMH heard during synchronization. The time to transmit a LMH's index is calculated based on the size of the static portion of the broadcast and the time of the dynamic portion multiplied by the time to broadcast one index item. This time is added to a LMH's transmit time and a SMH's receive time. In addition to the time hearing an index, each SMH hearing a LMH will calculate the amount of time needed to hear one static portion of the broadcast and add that to the SMH's receive time. Finally, the time to transmit any data items of interest will be added to a SMH's receive total. Any additional data push time remaining is added to a SMH's time in standby.

Each LMH calculates the size of the data broadcast as the size of the static portion and the number of data items remaining from the last data pull cycle. The time to transmit an index and broadcast of that size is calculated and added to the transmit time for each LMH. Any remaining data pull is added to the standby time for LMHs.

The time spent in data pull is added to the simulation time.

6.2.3.3 The Data Pull Stage Event

During data pull we first determine which nodes are active. This is done by calculating the distance between each node pair for all LMHs and SMHs and comparing this calculated distance to node transmission ranges. A SMH is inactive if no LMH and no SMHs are within the transmission range of the SMH. A LMH is inactive if no nodes were detected during synchronization. The next step is to calculate the load at each LMH. We do this by calculating which LMH is closest to each active SMH. This is done when calculating the distance between each SMH and each LMH. We then calculate the number of queries and routing requests each

LMH must serve, using data query and peer message frequencies. If insufficient time exists to handle each data query, the number of data queries not served is stored for the next data push stage. The amount of time each LMH and SMH spends in transmit, receive and standby mode is calculated and stored. The overall simulation time is then updated.

6.2.3.4 The Idle Stage Event

During the idle stage, we update the position of each LMH and SMH by allowing them to move a random and amount between *minMob* and *maxMob*. A random a direction and a random y direction is generated for each node. A random speed is also generated for each node, falling between *minMob* and *maxMob*. The distance in both the x and y direction is calculated by multiplying the random speed by the time since the last position update. However, the nodes are required to stay within the simulation region. If a resulting position is outside the roaming region, a node's position is on the edge of the region in the direction traveling.

Once new positions are determined for each LMH and SMH, the simulation time is compared to *period*. If period has been exceeded, the simulation moves to the reporting event and then terminates. If time remains, the simulation returns to the synchronization stage.

6.2.4 The Reporting Event

During the reporting event, all of the benchmark evaluation parameters (average power consumption SMH/LMH) percent of nodes not hearing broadcasts, broadcast effectiveness, peer efficiency and query efficiency) are calculated and stored to a file for later use. This is the final event in the simulation.

6.3 TriM Simulation Results

The proposed MANET data communication protocol was simulated for a variety of conditions within each of the three proposed scenarios. In Section 6.3.1, the results for the battlefield scenario are presented. Section 6.3.2 presents the results of the domestic rescue scenario. Finally, Section 6.3.3 presents the results of the simulation of the business scenario.

For each of the scenarios, simulation runs were completed for variations in the data broadcast, data query and peer-to-peer communications. We vary either the data broadcast size or the peer message/data query frequency with each simulation, but not both. First we vary the size of the broadcast. The benchmark includes three sizes. The frequency of data queries/peer messages are varied among the three benchmark settings.

We use the following terms for **broadcast size**:

- 50 items – Small Broadcast
- 100 items – Medium Broadcast
- 200 items – Large Broadcast.

The data pull stage consists of data queries and peer messaging. The frequency of both are always set to the same value during the simulation and are varied the same amount. Because they are varied together, we call this the pull frequency.

We use the following terms for **pull frequency**:

- 5 items/sec – Low Pull Frequency
- 20 items/sec – Medium Pull Frequency
- 40 items/sec – High Pull Frequency

For each of the nine variations, ten simulation runs are executed with the results averaged for each variation. The tables in Sections 6.3.1 to 6.3.3 are these averaged values. This is a total of 90 simulation runs per scenario. There will be 90 runs for the military battlefield scenario, 90 for the rescue scenario and 90 for the business scenario.

6.3.1 Battlefield Scenario Simulation Results

The first scenario simulated is the military battlefield. The overriding characteristic of this scenario is that there are only 20 LMHs with 1000 SMHs covering an area of 5 km x 5 km. In Table 5-1 it was estimated that the maximum area of broadcast transmission coverage is 2.6%. The simulation model assumes that the nodes are all initially in a small region 1 m x 1000 m. This is referred to as a staging area or battle line. An army does not just appear randomly in a theater of operation.

	Small Broadcast Low Pull Frequency	**Medium Broadcast Low Pull Frequency**	**Large Broadcast Low Pull Frequency**
SMH Average Power Consumption (watts)	0.23	0.13	0.13
LMH Average Power Consumption (watts)	8.10	7.09	6.69
Percent SMH Hearing Broadcast	21.36	17.29	15.73
Broadcast Effectiveness	85.26	86.33	83.13
Query Efficiency	7.45	7.86	9.67
Peer Efficiency	31.21	26.8	24.02

Table 6-3 Average Results for Battlefield Simulation – Low Pull Frequency

In Table 6-3 we see the average simulation results of benchmark parameters for the three broadcast sizes with a constant low request frequency for data queries and peer messages. It should be noted that when a LMH must choose between routing requests and serving data queries, routing takes precedence. The main rationale is that data queries can be added to the next data broadcast while peer messages are dropped at the end of data pull.

	Small Broadcast Medium Pull Frequency	**Medium Broadcast Medium Pull Frequency**	**Large Broadcast Medium Pull Frequency**
SMH Average Power Consumption (watts)	0.38	0.28	0.20
LMH Average Power Consumption (watts)	10.36	8.88	7.78
Percent SMH Hearing Broadcast	16.93	16.12	17.48
Broadcast Effectiveness	87.47	84.16	79.39
Query Efficiency	7.58	9.67	11.15
Peer Efficiency	33.29	33.2	30.52

Table 6-4 Average Results for Battlefield Simulation – Medium Pull Frequency

Table 6-4 shows the average simulation results of benchmark parameters for the three broadcast sizes with a medium level of data query and peer message frequency.

Table 6-5 shows the average simulation results of benchmark parameters for the three broadcast sizes and a high frequency rate for data queries and peer messages. Following the tables, the data is shown in chart form to allow us to see any trends in the data. Figure 6-2 shows the average SMH power consumption for all military scenario workloads. Figure 6-3 shows the average LMH power consumption. Figure 6-4 shows the percentage of SMHs hearing a broadcast. Figures 6-5, 6-6 and 6-7 show broadcast effectiveness, query effectiveness and peer effectiveness respectively.

	Small Broadcast High Pull Frequency	Medium Broadcast High Pull Frequency	Large Broadcast High Pull Frequency
SMH Average Power Consumption (watts)	0.45	0.31	0.27
LMH Average Power Consumption (watts)	11.49	9.27	9.08
Percent SMH Hearing Broadcast	16.22	15.01	19.27
Broadcast Effectiveness	85.51	84.96	75.99
Query Efficiency	9.51	9.07	13.12
Peer Efficiency	33.04	26.16	33.71

Table 6-5 Average Results for Battlefield Simulation – High Pull Frequency

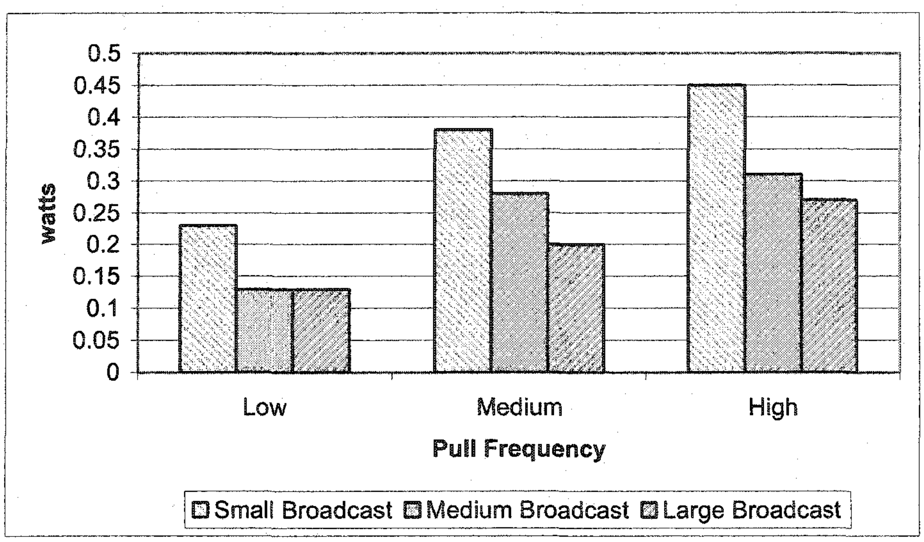


Figure 6-2 Average SMH Power Consumption Military Scenario

To understand the results, it is important to understand that the length of each stage of the service cycle changes from one scenario workload to the next. For instance, as the maximum size of the data broadcast increases, so does the length of the data push stage. When the pull frequency increases, the data push stage is also increased in length.

The average power consumption for SMHs is universally low, between receive and standby. This is the result of the high level of disconnection that occurs as the nodes move randomly in a very large area. As a large percentage of SMHs do not hear a LMH during synchronization, many SMHs spend a large percentage of their time in standby.

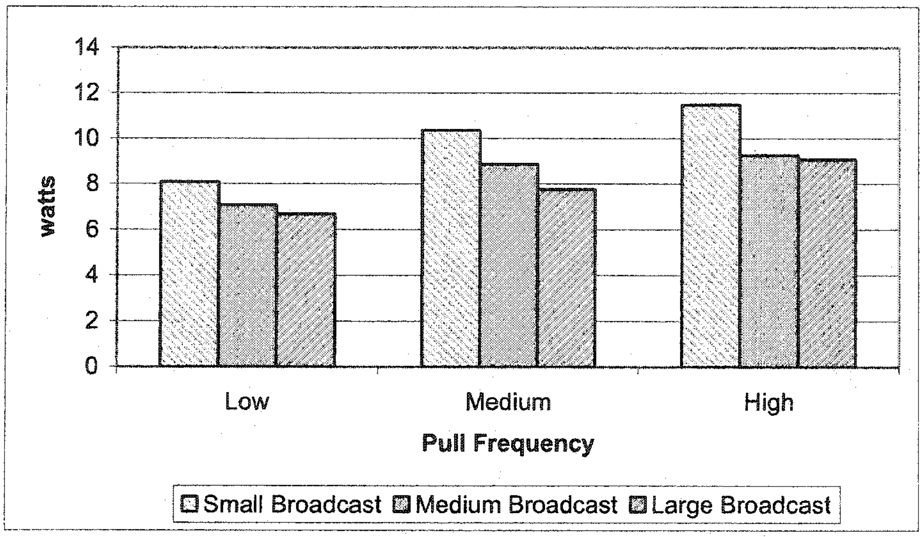


Figure 6-3 Average LMH Power Consumption Military Scenario

As the length of the service cycle increases because of a longer data pull corresponding to the larger data broadcast, the average power consumption for LMH decreases as less time is spent transmitting. While each node transmits longer, the amount of time waiting for all of the other nodes to transmit also increases. As the length of a service cycle increases due to larger data push caused by a larger pull frequency, the average power consumption increases due to the increase of transmission by each LMH. A larger pull frequency requires a greater number of data queries to be processed per unit time. Overall, the average power consumption for LMHs is near the level for receive mode. By infrequently transmitting, power consumption is kept low. LMHs spend most of their life in receive mode, waiting to serve data queries and routing requests. As the number of SMHs served by a LMH increases, the amount of time spent transmitting will also increase.

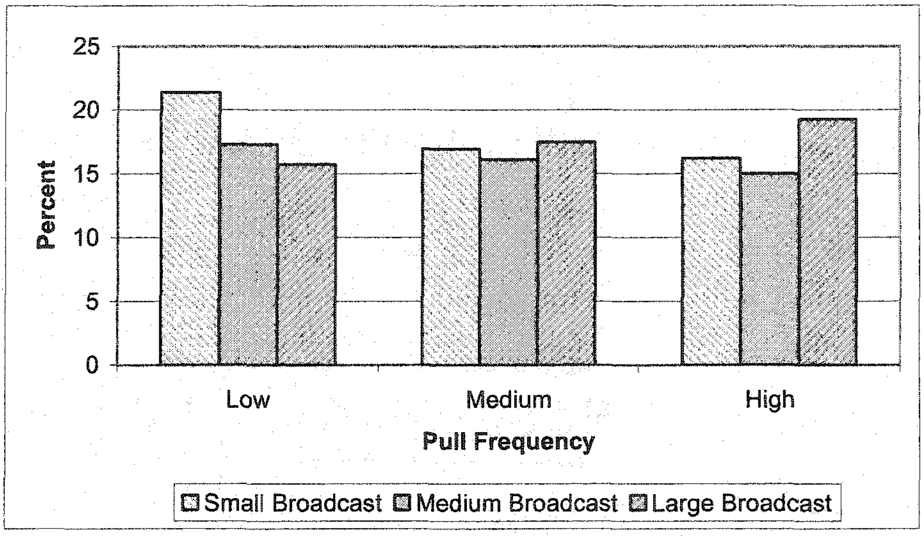


Figure 6-4 Percentage of SMHs hearing a Data Broadcast Military Scenario

The number of SMHs hearing a data broadcast is low. The number changes a small percent from one workload to another. The large size of the region, the random placement of LMHs and SMHs and the small number of LMHs makes the possibility of hearing a broadcast rather small.

In Figure 6-5 we see the simulated broadcast effectiveness. As expected, this number is very high. As the likelihood of a SMH hearing more than a single data broadcast is very small, the SMHs will not hear multiple static portions of broadcasts.

The broadcast effectiveness shown does decrease a small amount. As the pull frequency increases, the ability of a LMH to handle all data queries and peer message routing requests becomes slightly overwhelmed.

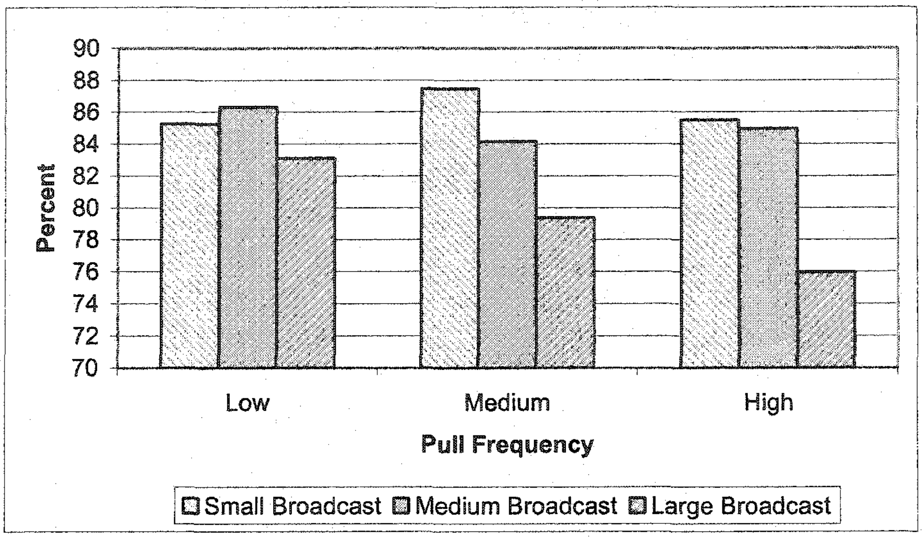


Figure 6-5 Broadcast Effectiveness
Military Scenario

The query efficiency and peer efficiency are both very low. The query efficiency is the lower of the two. This can be explained due to the large percentage of SMHs that are not near to a LMH. While 15 to 20% can hear a LMH data broadcast, the transmission radius of the SMH would mean that even fewer SMHs can be heard from a LMH for data query and message routing service. It should be noted that peer efficiency is much better. With the large number of SMHs in this scenario (1000) compared to LMH (20), a SMH is far more likely to be near a SMH with which to message. In addition, the simulation has a preference for peer message routing over query service when both are needed. Query efficiency is shown in Figure 6-6 and peer efficiency in Figure 6-7.

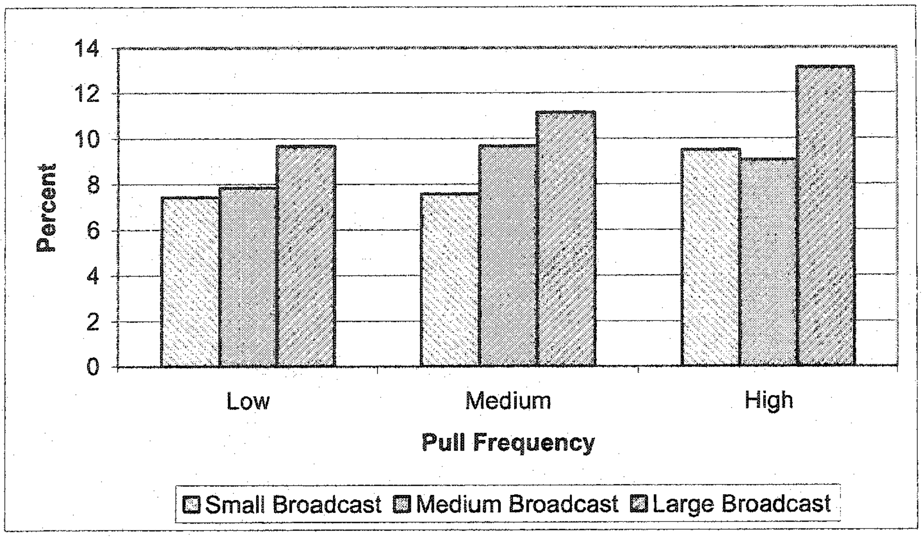


Figure 6-6 Query Efficiency
Military Scenario

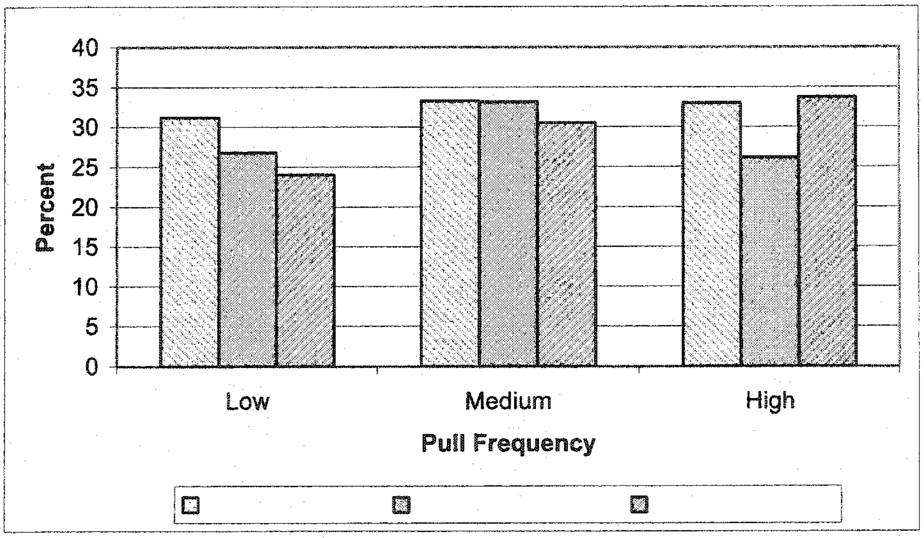


Figure 6-7 Peer Efficiency
Military Scenario

6.3.2 Domestic Rescue Scenario Simulation Results

In this section, the results of the domestic rescue simulation are presented. They are presented in the same order as the battlefield simulations. According to Table 5-1, the maximum coverage of broadcast transmission is 7.86%. In this scenario there are 10 LMHs and only 50 SMHs. The simulation region is a little smaller than the battlefield simulation at 5 km x 5 km.

	Small Broadcast Low Pull Frequency	**Medium Broadcast Low Pull Frequency**	**Large Broadcast Low Pull Frequency**
SMH Average Power Consumption (watts)	0.11	0.10	0.10
LMH Average Power Consumption (watts)	10.12	10.31	10.43
Percent SMH Hearing Broadcast	6.99	6.47	5.84
Broadcast Effectiveness	93.20	93.99	92.95
Query Efficiency	1.14	1.17	0.89
Peer Efficiency	2.94	2.52	2.38

Table 6-6 Average Results for Rescue Simulation – Low Pull Frequency

Table 6-6 shows the average simulation values for the three broadcast sizes with low request frequency. Table 6-7 shows the average simulation values for the three broadcast sizes with medium request frequency for peer messaging and data queries. Finally, Table 6-8 shows the average simulations values for the three broadcast sizes with high data query and peer frequency.

These three tables show much the same characteristics of the battlefield scenario. An increase in service cycle length due to data pull increases lowers average LMH power consumption while increases due to larger broadcasts increases average power consumption. Figures are not generated for this scenario and the business scenario as the same pattern is seen in the data.

It should be noted that broadcast effectiveness is higher and peer efficiency / query efficiency are lower in the rescue scenario than in the military scenario. While both use random placement and movement, the military initialization places LMHs and SMHs in closer proximity initially leading to higher values for these three parameters.

	Small Broadcast Medium Pull Frequency	**Medium Broadcast Medium Pull Frequency**	**Large Broadcast Medium Pull Frequency**
SMH Average Power Consumption (watts)	0.12	0.11	0.11
LMH Average Power Consumption (watts)	9.51	9.97	10.21
Percent SMH Hearing Broadcast	6.75	6.49	5.83
Broadcast Effectiveness	94.84	91.43	92.00
Query Efficiency	1.14	1.03	1.13
Peer Efficiency	3.40	3.28	3.23

Table 6-7 Average Results for Rescue Simulation – Medium Pull Frequency

As the average number of SMHs served by each LMH is low, the length of data pull, like in the battlefield scenario, is much shorter than the data push stage. This effectively keeps power consumption as low as possible for LMHs.

The percent of SMHs hearing a LMH broadcast is very close to the calculated maximum coverage shown in Table 5-1. This low percentage of connectedness leads to low query efficiency and low peer efficiency. These numbers were both slightly higher for the battlefield scenario. This is due to the larger number of SMHs near LMHs in the battlefield scenario.

	Small Broadcast High Pull Frequency	Medium Broadcast High Pull Frequency	Large Broadcast High Pull Frequency
SMH Average Power Consumption (watts)	0.12	0.11	0.11
LMH Average Power Consumption (watts)	8.87	9.53	10.00
Percent SMH Hearing Broadcast	7.42	6.69	5.69
Broadcast Effectiveness	97.42	92.73	90.57
Query Efficiency	1.23	1.20	1.16
Peer Efficiency	3.28	3.19	3.28

Table 6-8 Average Results for Rescue Simulation – High Pull Frequency

6.3.3 Business Scenario Simulation Results

Tables 6-9, 6-10 and 6-11, show the simulation results for the business scenario. In the business scenario we have a much smaller region, 1 km x 1 km. We also have fewer LMHs in the business scenario. The simulations used the maximum benchmark value of 6. The number of LMHs in the region was large at 1000 nodes. As in the rescue scenario, data is not shown in chart form as the data shows the same general pattern as in the military scenario.

	Small Broadcast Low Pull Frequency	Medium Broadcast Low Pull Frequency	Large Broadcast Low Pull Frequency
SMH Average Power Consumption (watts)	0.78	0.65	0.50
LMH Average Power Consumption (watts)	127.78	106.65	81.15
Percent SMH Hearing Broadcast	65.5	64.9	64.8
Broadcast Effectiveness	66.41	65.29	63.65
Query Efficiency	16.3	15.8	17.2
Peer Efficiency	49.86	49.37	47.89

Table 6-9 Average Results for Business Simulation – Low Pull Frequency

Table 6-9 shows the average results for the simulation with varying broadcast sizes and low query and peer message request frequencies. Table 6-10 shows the average simulation results for varying broadcast sizes with medium data query and peer message frequencies. Table 6-11 shows the average simulation results for the three broadcast sizes with a high peer message and data query frequency. Several things are immediately apparent. Query efficiency is up some. Both peer efficiency and LMH power consumption are much higher than the previous scenarios.

	Small Broadcast Medium Pull Frequency	Medium Broadcast Medium Pull Frequency	Large Broadcast Medium Pull Frequency
SMH Average Power Consumption (watts)	0.96	0.88	0.78
LMH Average Power Consumption (watts)	156.35	145.21	132.80
Percent SMH Hearing Broadcast	66.2	65.9	66.0
Broadcast Effectiveness	67.37	65.45	63.41
Query Efficiency	16.2	16.9	16.7
Peer Efficiency	58.0	57.4	56.9

Table 6-10 Average Results for Business Simulation – Medium Pull Frequency

In this scenario, unlike the other two, the level of disconnection in the MANET is low as a large number of nodes occupy a small space. In fact, the 6 LMHs serve a population of 1000 SMHs. Because of this, the amount of time spent in data pull is much higher than the time spent in data push. Data pull is more expensive as each LMH responds to individual queries and routing requests. During data pull, LMHs serving a large number of SMHs may spend the majority of their time in transmit mode.

The average power consumption for SMHs is nearly at the receive level. Nearly two thirds of the SMHs were within the reach of a LMH transmission. The broadcast effectiveness was down some as a SMH was more likely to be within range of more than 1 LMH. The static portion of the broadcast transmission is then duplicative and lowers broadcast effectiveness. While effectiveness is lower, far more SMHs are served.

	Small Broadcast High Pull Frequency	Medium Broadcast High Pull Frequency	Large Broadcast High Pull Frequency
SMH Average Power Consumption (watts)	1.01	0.96	0.91
LMH Average Power Consumption (watts)	163.15	157.63	147.20
Percent SMH Hearing Broadcast	65.7	66.1	65.5
Broadcast Effectiveness	66.45	64.13	65.27
Query Efficiency	17.1	16.8	17.3
Peer Efficiency	57.8	58.1	57.3

Table 6-11 Average Results for Business Simulation – High Pull Frequency

Query efficiency is still rather low. This can be accounted for in two ways. First, less than two thirds of the SMHs cannot make a data request. While two thirds heard a broadcast, the transmission range of a SMH is less than half of the transmission range of a LMH. Second, the large number of SMH served by each of the very few LMHs will be high. Serving peer routing requests and the large number of queries served decreases the likelihood of a successful data query being served.

Table 5-1 calculates the maximum broadcast transmission coverage at 118%. This would seem to imply that all SMHs would be able to hear a transmission. However, if a LMH is near a region boundary, a portion of the transmission reaches outside of the region, decreasing region coverage. An actual coverage of approximately two thirds is not unreasonable or unexpected.

Perhaps the most interesting result from the simulation is the large number of similar results, regardless of broadcast size or query/peer frequencies. These values are more a function of LMH coverage and region size than length of any stage in the service cycle. As the coverage remains the same and very high, the effectiveness of peer messaging and data queries should be very similar.

6.3.4 Other Simulation Results

One thing noted in the three benchmark scenarios simulated is that the number of SMHs within reach of a LMH never gets near 100% and Peer Efficiency and Query Efficiency remains low. It is also useful to know how the simulation model performs for values outside of the expected range of benchmark values.

In this section, the number of LMHs and transmission range are varied in the simulation to see on the benchmark evaluation parameters. We also observe the behavior of the simulation with unanticipated values. In all of these simulations we use a medium sized broadcast with medium data query and peer messaging frequencies. In Tables 6-12, 6-13, and 6-14, the number of LMHs is varied for the battlefield, rescue and business scenarios respectively.

In Tables 6-15, 6-16 and 6-17 the broadcast ranges of LMHs and SMHs is varied for the battlefield, rescue and business scenarios respectively. In each of these, the SMH radius is half of the LMH Radius except in the benchmark 250 meter case where the SMH radius is 100 meters. Figure 6-7 shows the percent of SMHs hearing a LMH broadcast for varied numbers of LMHs while Figure 6-8 and 6-9 show query efficiency and peer efficiency for varied numbers of LMHs respectively. Figure 6-10, 6-11 and 6-12 show the same three values, but for varied transmission rages.

	20 LMHs	50 LMHs	100 LMHs	500 LMHs	1000 LMHs
SMH Average Power Consumption	0.28	0.14	0.13	0.14	0.18
LMH Average Power Consumption	8.88	4.52	3.43	2.93	3.72
Percent SMH Hearing Broadcast	16.01	29.42	40.68	96.00	99.75
Broadcast Effectiveness	82.81	63.83	33.29	1.22	0.41
Query Efficiency	9.83	17.04	32.01	87.33	95.56
Peer Efficiency	33.36	34.55	49.88	94.72	98.90

Table 6-12 Battlefield Simulation – Varied LMHs
Medium Broadcast / Medium Pull Frequency

The results in Table 6-12 look promising. With 500 or 1000 LMHs we get a very high percentage of SMHs hearing a broadcast, having queries processed and peer messages sent and received. In addition, the power consumption rate for LMHs gets very low as a large percentage of time is spent in standby while other LMHs transmit data broadcasts. There is a cost associated with this. Each server is an investment. If too many servers are put into service, the cost in time and money can be substantial. In addition, the length of the service cycle increases dramatically as the protocol specifies sequential transmission of data broadcasts. Clearly the number of transmission heard by each SMH is also large as the broadcast effectiveness becomes quite small.

In Table 6-13 the simulations were only run to 50 LMHs. In this scenario, there are only 50 SMHs. Having more servers than clients does not seem reasonable. The number of nodes in the 5km x 5 km region remains quite small and while the benchmark parameters increase positively, the percentage of SMHs of hearing a

LMH broadcast, peer efficiency and query efficiency remains rather low.

	10 LMHs	20 LMHs	50 LMHs
SMH Average Power Consumption	0.11	0.11	0.11
LMH Average Power Consumption	9.97	6.29	3.81
Percent SMH Hearing Broadcast	6.83	12.00	34.26
Broadcast Effectiveness	91.64	93.42	81.60
Query Efficiency	1.03	2.64	11.22
Peer Efficiency	3.38	4.66	10.22

Table 6-13 Rescue Simulation – Varied LMHs
Medium Broadcast / Medium Pull Frequency

	6 LMHs	10 LMHs	20 LMHs	50 LMHs	100 LMHs
SMH Average Power Consumption	0.88	00.73	0.43	0.18	0.14
LMH Average Power Consumption	145.21	143.01	137.98	130.74	128.85
Percent SMH Hearing Broadcast	65.6	77.10	91.70	100	100
Broadcast Effectiveness	66.33	50.69	28.35	12.36	6.30
Query Efficiency	16.40	23.60	46.30	74.80	93.30
Peer Efficiency	58.2	61.80	73.15	87.40	96.65

Table 6-14 Business Simulation – Varied LMHs
Medium Broadcast / Medium Pull Frequency

Table 6-14 shows an interesting case. The number of SMHs in the small business region is large, 1000. As we increase the number of LMHs, query efficiency and peer efficiency increases. However, broadcast effectiveness is extremely low, indicating a lot of wasted broadcast transmission, as a SMH will hear several LMHs.

As with the other business scenarios, the power consumption of LMHs remains high as the number of broadcast transmissions increase. However, data pull dominates this scenario with the large number of SMHs served by each LMH.

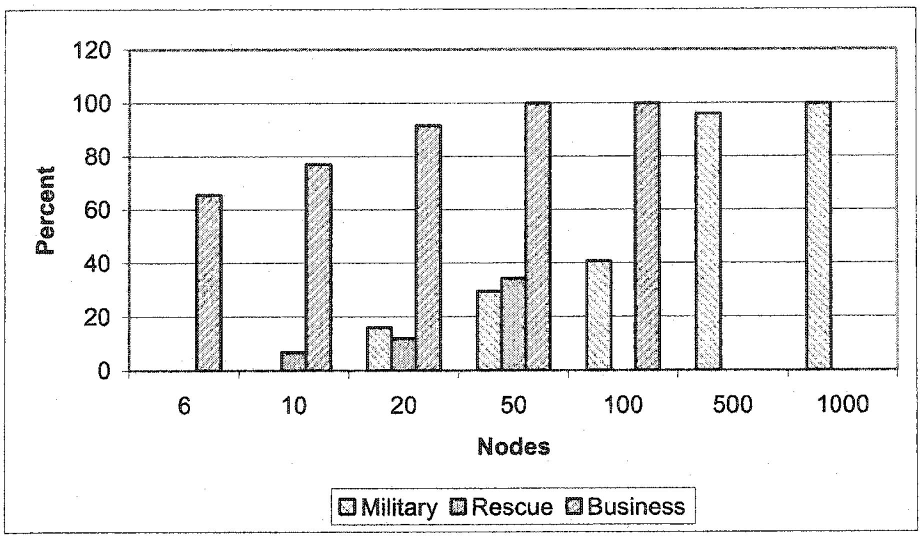


Figure 6-8 Percent Hearing Broadcast – Varying Number LMHs

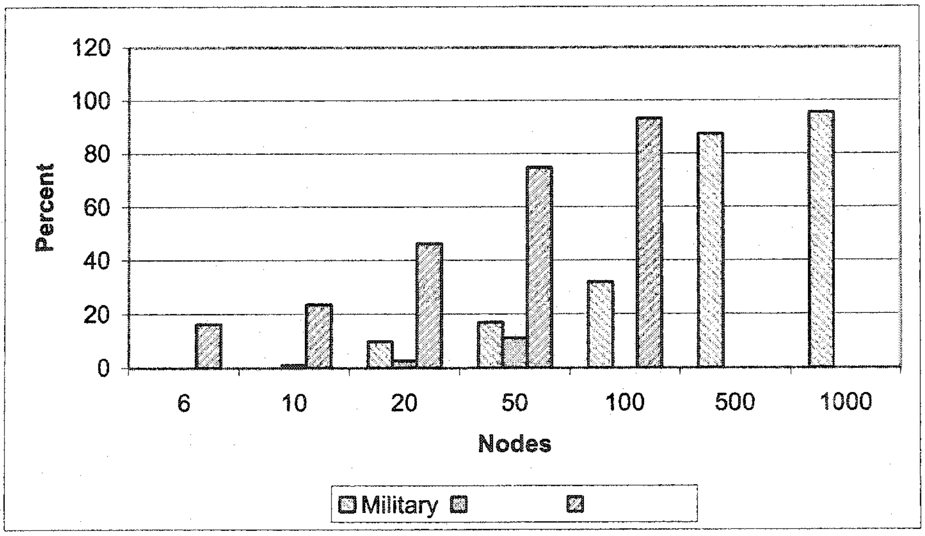


Figure 6-9 Query Efficiency – Varying Number LMHs

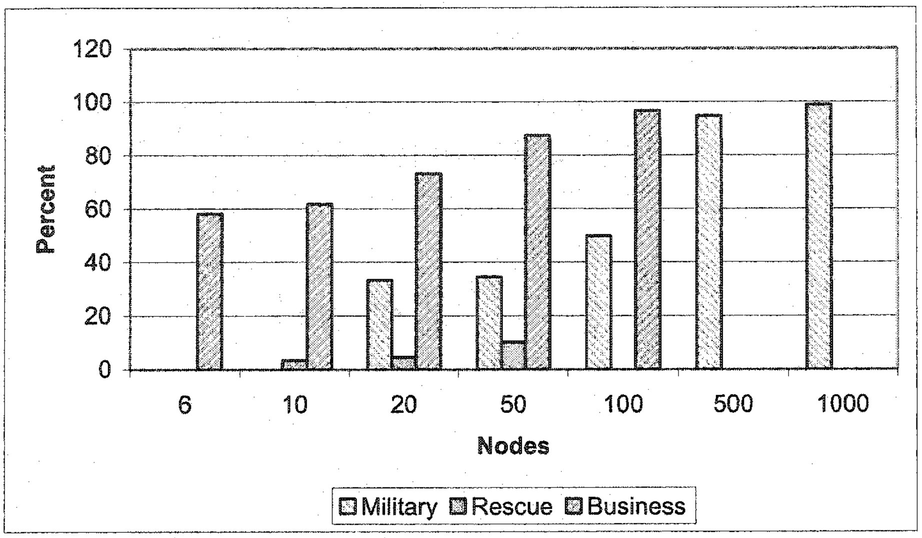


Figure 6-10 Peer Efficiency – Varying Number LMHs

Figures 6-8, 6-9 and 6-10 provide a perspective on the effect of adding LMHs into a scenario. The behavior of the simulation in these cases was as expected. One thing to note is that to get excellent peer and query efficiencies requires a certain level of SMH saturation. If most of the SMHs cannot reach a LMH, these numbers will remain small.

In the following tables we see the effect of increasing transmission ranges for LMHs and SMHs. Each scenario is simulated with a medium sized broadcast and a medium frequency of data queries and peer messages. In the LMH 250 m. case, the SMH range is 100 m. For the other cases, the SMH range is 50% of the LMH transmission ranges. The purpose of simulating varying transmission ranges is to see the effect of technological improvements. The benchmark values for number of LMHs and SMHs are used. As with changing numbers of LMHs, we will chart percent hearing broadcast, query efficiency and peer efficiency. These will be Figures 6-11, 6-12 and 6-13 respectively.

Transmission Ranges	LMH 250 m SMH 100 m	LMH 500 m SMH 250m	LMH 1000 m SMH 500 m	LMH 2000 m SMH 1000 m
SMH Average Power Consumption	0.28	0.37	0.41	0.42
LMH Average Power Consumption	8.88	11.04	59.76	61.03
Percent SMH Hearing Broadcast	16.37	32.97	59.16	87.26
Broadcast Effectiveness	87.31	76.26	53.86	24.80
Query Efficiency	9.31	16.37	32.97	60.69
Peer Efficiency	32.84	51.48	65.34	80.16

Table 6-15 Battlefield Simulation – Varied Transmission Ranges Medium Broadcast / Medium Pull Frequency

Table 6-15 shows the results for a simulation using 20 LMHs and 1000 SMHs in a 10 km x 15 km region. The transmission ranges are varied, as specified. Table 6-16 shows the results for simulation in the domestic rescue scenario with 10 LMHs and 50 SMHs in a 5 km x 5 km region.

Transmission Ranges	LMH 250 m SMH 100 m	LMH 500 m SMH 250m	LMH 1000 m SMH 500 m	LMH 2000 m SMH 1000 m
SMH Average Power Consumption	0.11	0.14	0.20	0.24
LMH Average Power Consumption	9.97	10.63	11.54	12.14
Percent SMH Hearing Broadcast	6.83	23.75	60.76	94.19
Broadcast Effectiveness	91.64	87.14	62.62	29.68
Query Efficiency	1.03	6.83	23.75	60.76
Peer Efficiency	3.38	17.52	48.62	80.14

Table 6-16 Rescue Simulation – Varied Transmission Ranges Medium Broadcast / Medium Pull Frequency

Table 6-17 shows the results for a simulation using 6 LMHs and 1000 SMHs in a

1 km x 1 km region. The transmission ranges are varied, as specified.

Transmission Ranges	LMH 250 m SMH 100 m	LMH 500 m SMH 250m	LMH 1000 m SMH 500 m	LMH 2000 m SMH 1000 m
SMH Average Power Consumption	0.88	0.89	0.89	0.89
LMH Average Power Consumption	145.21	150.35	154.12	154.65
Percent SMH Hearing Broadcast	65.60	97.20	100	100
Broadcast Effectiveness	66.33	30.43	16.86	16.67
Query Efficiency	16.40	65.60	97.20	100
Peer Efficiency	58.20	82.80	98.60	100

Table 6-17 Business Simulation – Varied Transmission Ranges Medium Broadcast / Medium Pull Frequency

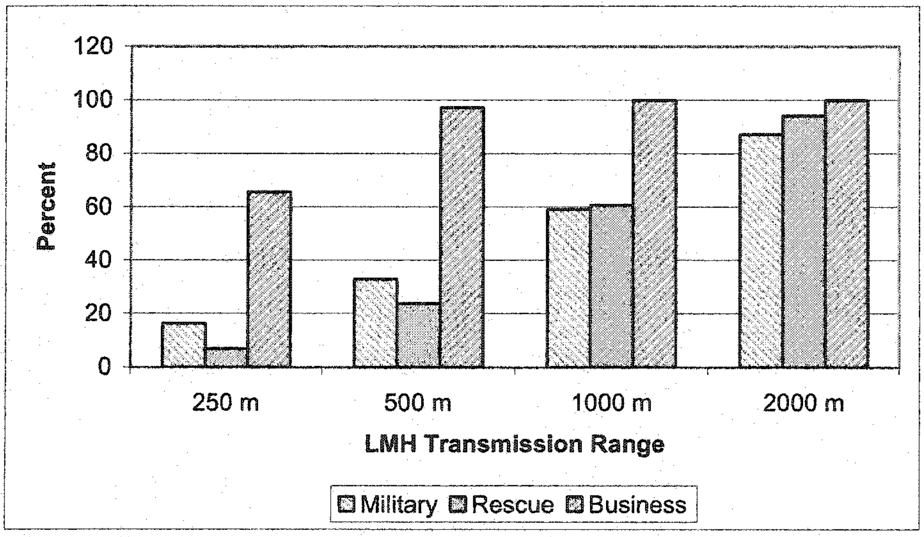


Figure 6-11 % SMHs Hearing Broadcast – Varying Transmission Distance

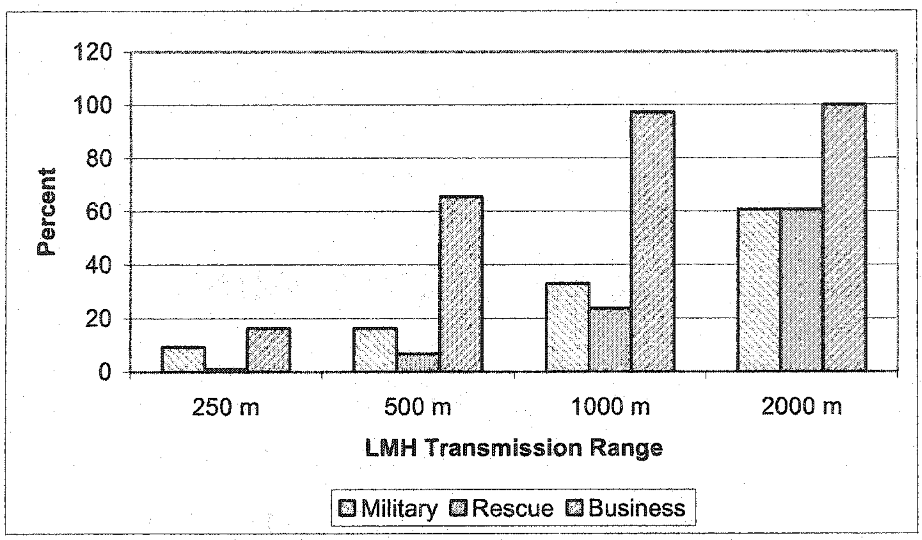


Figure 6-12 Query Efficiency – Varying Transmission Distance

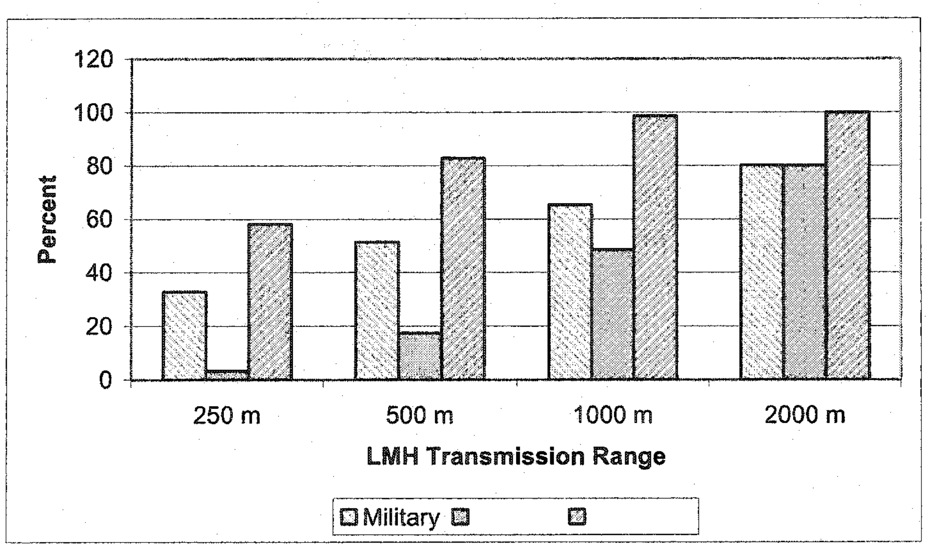


Figure 6-13 Peer Efficiency – Varying Transmission Distance

The varied transmission ranges appear to be a very good option. In each scenario, the average power consumption did not increase significantly. However, it is expected that a longer transmission would have a greater power cost. What that cost would be is unknown, so the power dissipation rates of the benchmark were used. It should be noted that increasing the transmission range of a LMH or SMH does not increase the length of the service cycle or the amount of time spent in any of the various power modes.

This shows is that increasing the transmission distance for LMHs and SMHs is one way to improve the performance of a MANET with regard to SMHs hearing the broadcast, query efficiency and peer efficiency without deploying additional LMHs.

6.4 Summary of Simulation

In this chapter, the simulation plan is explained. First we explain the basic structure of the simulation model and assumptions made. This model directly reflects the structure of the proposed MANET data communication protocol's four stages. Next the assumptions made in setting up the simulation are explained in detail. A simulation is not an actual MANET and so various assumptions in how the simulation will proceed and how nodes will behave must be made.

Following the explanation of the protocol design, the results of 9 workloads for each of three scenarios are presented. Some initial analysis in explanation of these results is also made. The simulation showed the behavior of the proposed protocol under benchmark conditions assuming random distribution of nodes in the simulation region. The validity of this assumption is further explored in Chapter 7. In addition to the simulations related to the benchmark of Chapter 4, additional simulations varying the number of LMHs and the transmission ranges of LMHs were performed and the results presented in Section 6.3.4.

The simulations showed that with respect to power consumption, the protocol performed very well when the network was sparsely populated. In particular, in the battlefield and rescue scenarios, power consumption remained low. This appears to have occurred because the proposed protocol is able to determine when tasks requiring higher expenditure of energy are unnecessary. In addition, the behavior of the protocol allows coordinated effort by all nodes, reducing network collisions and the result retransmission that collisions entail.

The simulations also showed that in environments where a large number of SMHs have their data needs met by a small number of LMHs, data query becomes less energy-efficient and the cost to LMHs is greatly increased. Finally, the simulations showed that in sparsely populated regions, as represented by the military and rescue scenarios, network performance can be increased markedly through increasing the transmission range of LMHs and SMHs.

These results will be compared to the work of Gruenwald [37] and Wieselthier [72] in Chapter 7. Also in Chapter 7, the simulation results will be compared to the expected results, calculated using the evaluation formulas of Chapter 5. Final conclusions will then be drawn with directions for future work.

Chapter 7

MANET DATA COMMUNICATION PROTOCOL CONCLUSIONS

This research has proposed a new MANET data communication protocol, TriM. TriM was proposed after examining the issues associated with MANET data communication and the scenarios in which these ad-hoc networks are useful. The research in MANET data communication and related areas was examined. This examination pointed out weaknesses in current MANET data communication protocols. The review of the literature also pointed out weaknesses in the evaluation of competing protocols. TriM, was presented in Chapter 3.

To address the weaknesses in MANET data communication protocol evaluation, a benchmark was developed and presented in Chapter 4. The benchmark was used in the evaluation of the network. The benchmark provided guidance in both the analysis and simulation of the proposed protocol. The simulation showed the performance of the network under a range of different scenarios and workloads required by the benchmark. The MANET was also simulated under non-benchmark conditions. The evaluation criteria provided by the benchmark allowed the evaluation of the protocol under both the standard and non-standard conditions. This benchmark serves as a starting point in the dialog that must take place within the MANET data communication community.

After presenting the new protocol and benchmark it was evaluated in three ways. First, the MANET data communication issues presented in Chapter 1 were compared to the proposed protocol. The manner in which the protocol addresses those issues was discussed in Chapter 5. Also in Chapter 5, the expected performance of the protocol was defined mathematically according to the benchmark evaluation criteria. In this evaluation, formulas for the expected performance of the protocol were developed for average SMH/LMH power consumption, average percent of SMHs able to hear a broadcast, broadcast effectiveness, query efficiency and peer efficiency. Finally, the protocol was simulated using the developed benchmark for the scenarios described in the research. These scenarios are military battlefield, domestic rescue and temporary business networks.

In this Chapter the evaluation of the protocol is brought together. In Section 7.1, the results of the simulation in Chapter 6 are compared to the mathematical description of the protocol developed in Chapter 5. In addition to presenting these results, they are compared and discrepancies are examined.

Section 7.2 compares the proposed protocol to the MANET data communication protocols most similar to the proposed protocol. The protocols used for evaluation are the protocols of Wieselthier [72] and Gruenwald [37]. These protocols do not provide all three methods of MANET data communication, so the comparison cannot be complete. However, similarities and differences between the results of the proposed protocol with the published results of these protocols will be discussed.

Section 7.3 presents the final conclusions of this research. Section 7.4 suggests future research directions for the proposed benchmark and protocol. Following Chapter 7 are the bibliography and appendices. Appendix A lists all variables used in TriM, the proposed benchmark and in benchmark evaluation. These are listed alphabetically with reference to where they are used in the benchmark, protocol or evaluation. This appendix allows the reader to see the common values used across all three areas. Appendix B contains the C insert code used with the simulation of the proposed protocol.

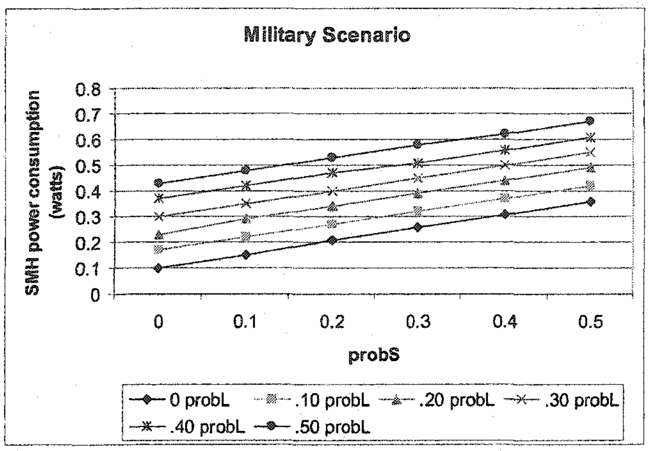
7.1 Comparison of Protocol Simulation and Mathematical Description

In Chapter 5, a group of equations were developed to allow the approximation of the performance of the proposed MANET data communication protocol prior to simulation. These were for average power consumption of the SMH and LMH, percent of broadcast coverage, broadcast effectiveness, query efficiency and peer efficiency. We discuss each in order. For each parameter an expected value was calculated. Where equations included probabilities, several values were calculated for a range of possible probabilities. These values are then compared to the simulation results

7.1.1 Average SMH Power Consumption Comparison

Figure 7-1 shows the range of average power consumption values for a variety of probabilities. Each line in the chart represents the probability that a LMH was detected during synchronization for a particular scenario. Each point on the line represents the probability that a SMH was detected during synchronization. To the side, the simulation values are shown for the military scenario.

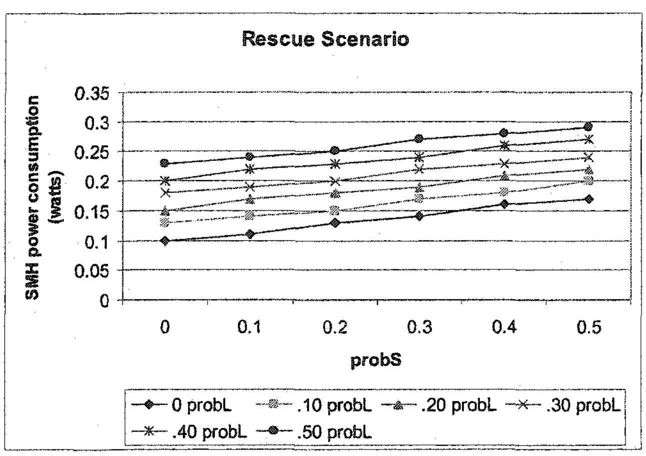
Figure 7-2 shows the same information for the rescue scenario and Figure 7-3 for the business scenario. For both the military and rescue scenarios, the density of LMHs and SMHs is very small. As expected, the average SMH power consumption is towards the far left of the graph while the business scenario with a higher node density has values in the appropriate range on the right portion of the graph.



LMH	Density	2.60%
SMH	Density	20.90%

	Low Pull	Med Pull	High Pull
Small Broadcast	0.23	0.38	0.45
Medium Broadcast	0.13	0.28	0.31
Large Broadcast	0.13	0.2	0.27

Figure 7-1 Simulated and Calculated Average SMH Power Consumption Comparison in Military Scenario

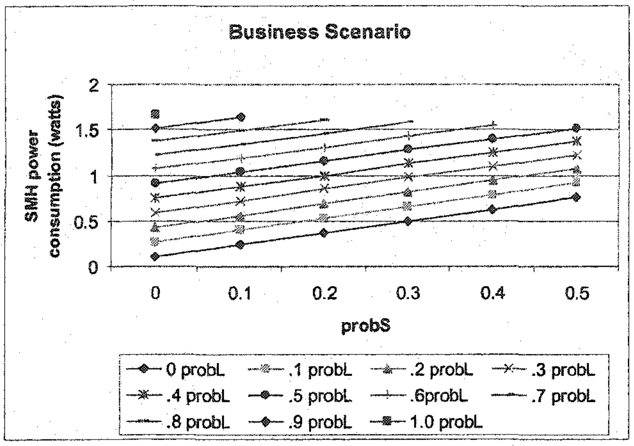


LMH	Density	7.85%
SMH	Density	6.30%

	Low Pull	Med Pull	High Pull
Small Broadcast	0.11	0.12	0.12
Medium Broadcast	0.1	0.11	0.11
Large Broadcast	0.1	0.11	0.11

Figure 7-2 Simulated and Calculated Average SMH Power Consumption Comparison in Rescue Scenario

Figures 7-1, 7-2 and 7-3 show the comparison between the simulated and calculated values in all three scenarios, the average SMH power consumption values from the simulation fell on the chart in the area where the *probL* and *probS* were anticipated due to the density of each scenario.



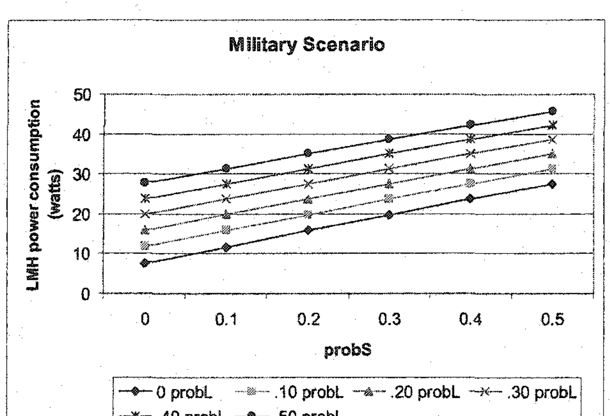
LMH	Density	100.00%
SMH	Density	100.00%

	Low Pull	Med Pull	High Pull
Small Broadcast	0.78	0.96	1.01
Medium Broadcast	0.65	0.88	0.96
Large Broadcast	0.5	0.78	0.91

Figure 7-3 Simulated and Calculated Average SMH Power Consumption Comparison in Business Scenario

7.1.2 Average LMH Power Consumption Comparison

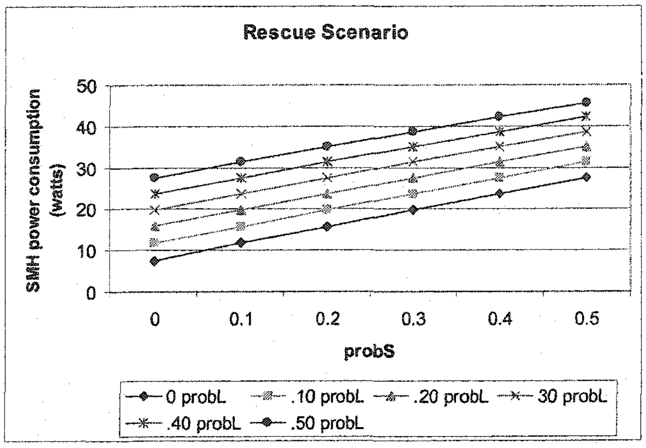
Figures 7-4, 7-5 and 7-6 show the comparison between the simulated and calculated average LMH power consumption rates. Both the SMH and LMH calculated value is a maximum value as a broadcast of maximum size is assumed. It is expected that simulation values will generally be a lower, as smaller broadcasts often occur during simulation. Figure 7-4 is for the military scenario, Figure 7-5 for rescue and Figure 7-6 for business.



LMH	Density	2.60%
SMH	Density	20.90%

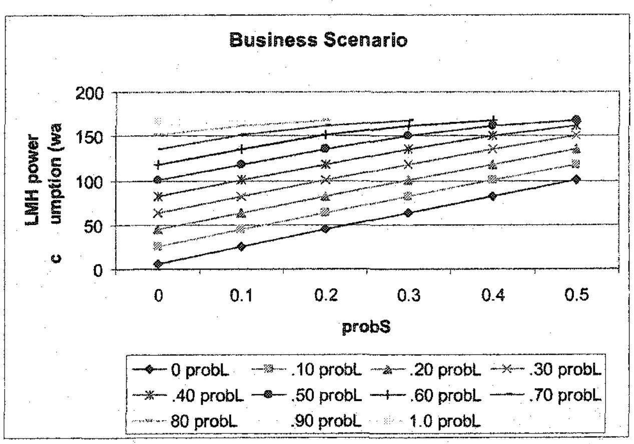
	Low Pull	Med Pull	High Pull
Small Broadcast	8.1	10.36	11.49
Medium Broadcast	7.09	8.88	9.27
Large Broadcast	6.69	7.78	9.08

Figure 7-4 Simulated and Calculated Average LMH Power Consumption Comparison in Military Scenario



LMH	Density	7.85%	
SMH	Density	6.30%	
	Low Pull	Med Pull	High Pull
Small Broadcast	10.12	9.51	8.87
Medium Broadcast	10.31	9.97	9.53
Large Broadcast	10.43	10.21	10

Figure 7-5 Simulated and Calculated Average LMH Power Consumption Comparison in Rescue Scenario



LMH	Density	100.00%	
SMH	Density	100.00%	
	Low Pull	Med Pull	High Pull
Small Broadcast	127.78	156.35	163.15
Medium Broadcast	106.65	145.21	157.63
Large Broadcast	81.15	132.8	147.2

Figure 7-6 Simulated and Calculated Average LMH Power Consumption Comparison in Business Scenario

7.1.3 Average Percent Hearing Broadcast Comparison

In the military and rescue scenarios, it is anticipated that few will hear the data broadcasts as the broadcast coverage of the LMHs as a percentage of the roaming region is very small. In Figure 7-7 we see the average coverage for the military scenario. Two simulated values are given. The stage value is for simulations where all nodes start randomly in the smaller staging area. In the random simulation, nodes start randomly throughout the region. The benchmark assumes random placement of nodes. Figure 7-8 and 7-9 show the comparisons for the rescue and business scenarios respectively.

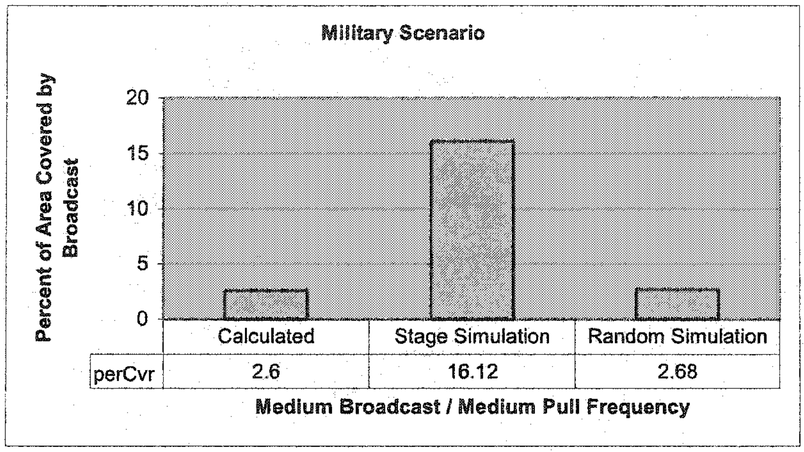


Figure 7-7 Simulated and Calculated Percent Hearing Data Broadcast Comparison in Military Scenario

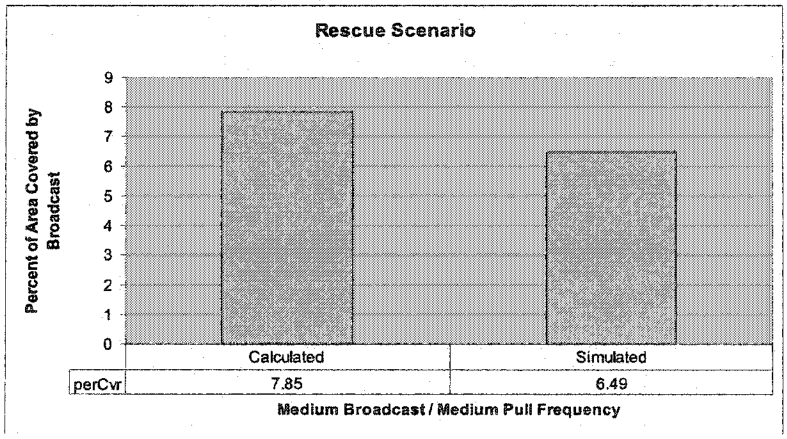


Figure 7-8 Simulated and Calculated Percent Hearing Data Broadcast Comparison in Business Scenario

As with the average power consumption, the values calculated for TriM and the TriM simulation values are very close to the same. The one notable difference is the business environment. The predicted value anticipated full coverage, which did not occur in simulation.

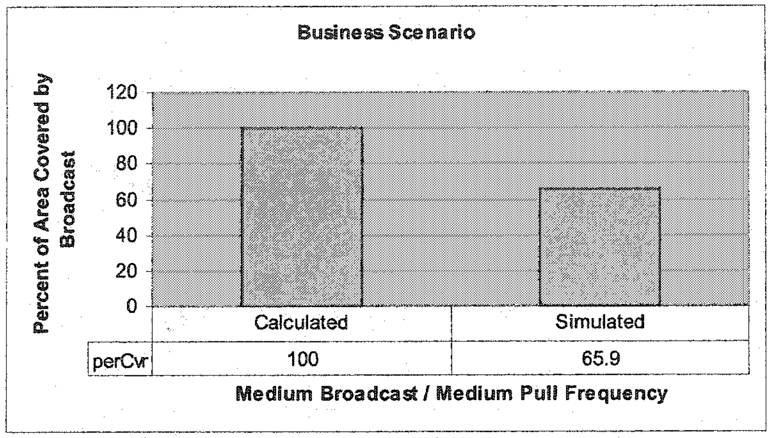
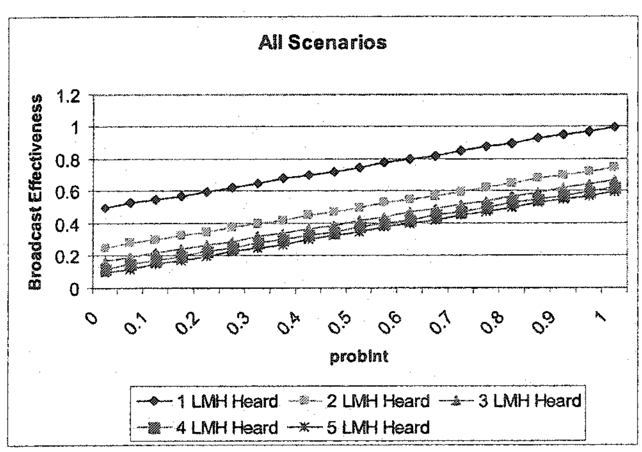


Figure 7-9 Simulated and Calculated Percent Hearing Data Broadcast Comparison in Business Scenario

7.1.4 Broadcast Effectiveness Comparison

Broadcast effectiveness is a measure of the usefulness of broadcasts heard. As the *probInt* (the probability a dynamic item is of interest) was set to 1 in the simulation, a lower broadcast effectiveness indicates that a SMH heard multiple broadcasts. Only the duplicative static portion of the broadcast would lower broadcast effectiveness.



Medium Broadcast - Medium Pull		
Mil. Staged	Rescue	Business
0.84	0.91	0.65

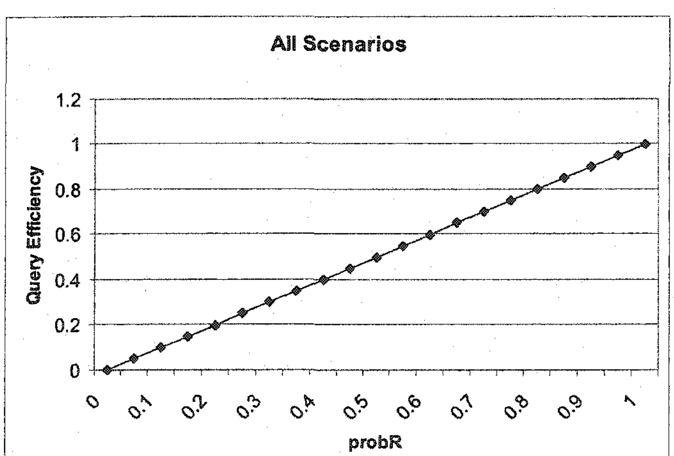
Mil. Random
0.98

Figure 7-10 Simulated and Calculated Broadcast Effectiveness Comparison

We anticipate that in the military and rescue scenarios, only a single broadcast will normally be heard. In the business scenario, we anticipate that 2 or more broadcasts will be within range. This is what we observe in Figure 7-10.

7.1.5 Query Efficiency Comparison

The maximum query efficiency is obtained when every SMH is close enough to a LMH to request data service. However, as Figure 7-11 shows, the percentage of area within 100 meters of a LMH is small. This is the transmission range of a SMH. If the SMH is further away, the LMH will not hear its transmission. The area covered in the military scenario is less than 1%. The rescue scenario is slightly larger but still less than 2%. Even in the business scenario, only a maximum of 19% of the region can query a LMH. The simulated query efficiencies are very close to the maximum possible assuming a random distribution of nodes. *probR* is the probability a SMH can reach a LMH for data query.



Medium Broadcast - Medium Pull		
Mil. Staged	Rescue	Business
0.097	0.01	0.169

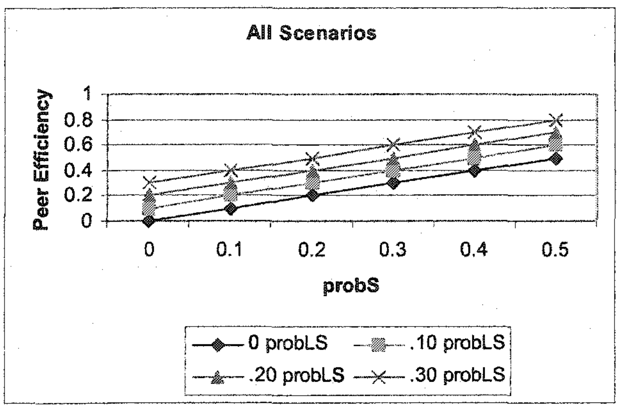
Mil. Random
0.007

Area Within 100 meters of LMH		
0.42%	1.30%	19%

Figure 7-11 Simulated and Calculated Query Efficiency Comparison

7.1.6 Peer Efficiency Comparison

Peer efficiency measures the ability to send and receive peer messages. Peer efficiency tends to be better than query efficiency as a SMH may also send peer messages to SMHs detected even if no LMHs were detected. Figure 7-12 shows the comparison between the calculated and simulated peer efficiency values.



Medium Broadcast - Medium Pull		
Mil. Staged	Rescue	Business
0.33	0.03	0.57

Mil. Random
0.13

Assumes:
50% of peer messages are multi-hop
50% of peer messages are single hop

Figure 7-12 Simulated and Calculated Peer Efficiency Comparison

The probability a SMH detects both a SMH and LMH during synchronization is represented by the lines. The points are the probability that only a SMH was detected. Again, due to its higher density of nodes, the business scenario has a much better peer efficiency.

The predicted values and the simulated values in all cases were very close. This gives confidence in the benchmark's ability to predict the performance of a MANET data communication protocol. These values can be verified by simulation. Our confidence in the simulation is also higher due to the similar results.

The nature of a sparsely populated network makes data query and peer messaging less important as a large percentage of the nodes are not sufficiently close to other nodes to make use of either capability.

7.2 Comparison to Previous MANET Data Communication Protocols

The most relevant two MANET data communication protocols to the work presented in this research are the protocols of Gruenwald [37] and Wieselthier [70][71][72]. In this section the protocol proposed in this research will be compared to these two protocols.

7.2.1 Comparison to Protocol of Gruenwald, Gu and Javed

The Leader Selection protocol of [37] is a soft real-time MANET data broadcast protocol. Data query, as described in the proposed protocol does not exist. Rather, data requests help inform the building of subsequent data broadcasts. Individual data items are not served interactively.

The protocol of [37] provides 4 measures for evaluation. These are: energy consumed by LMHs, energy consumed by SMHs, access time and broadcast hit ratio. A complete comparison cannot be made, however a partial comparison is possible. TriM does not calculate access time. However, energy consumption is measured for both LMHs and SMHs. In addition, broadcast effectiveness is similar to the broadcast hit ratio of Gruenwald when the probability that dynamic items in the broadcast are of interest is 1.

The parameters specified in [37] are similar to but not identical to any of the benchmark scenarios. To make the comparison as accurate as possible, the simulation was rerun using as many of the parameters of [37] as possible, including number of SMHs and LMHs, CPU power, bandwidth, and transmission radius, size of simulation region and database size. Each LMH transmitted 20% of the database in each data broadcast, the percent of hot items in Gruenwald's database.

Table 7-1 shows the comparison between the Gruenwald's protocol and the protocol described in this research. Peer efficiency and Query efficiency were not calculated, as they had no corresponding value in [37].

	TriM	Leader Selection
SMH Average Power Consumption (watts/hr)	0.19	20-60
LMH Average Power Consumption (watts/hr)	18.99	15-24
Percent SMH Hearing Broadcast	95.9	not applicable
Broadcast Effectiveness (Broadcast Hit Ratio)	70.36 BE	60-100 BHR

Table 7-1 TriM Comparison to Leader Selection Protocol

The behavior of TriM was similar to the Leader Selection protocol. The major departure is in the SMH power consumption. This is due to a difference in how SMHs are used in the 2 protocols. In Leader Selection, SMHs drive the contents of the data broadcast. In TriM, the SMHs only request what was not received in a recent data broadcast.

The primary advantage of TriM or Leader Selection is the addition of peer messaging and interactive data query. These items are not available in Leader Selection. TriM compares favorably with Leader Selection. As the SMHs are used very differently, it is quite possible that the different protocols will be better suited to different MANET scenarios.

7.2.2 Comparison to Protocol of Wieselthier, Nguyen and Ephremides

The protocol of Wieselthier cannot be directly compared to TriM. Wieselthier provides insufficient details to fully replicate his work by simulation comparison. However, some comparison is possible. Wieselthier assumes stationary nodes. This certainly simplifies the problem, but is not realistic in a MANET environment. By comparison, the proposed protocol assumes node mobility and was designed with that consideration in mind.

Wieselthier also builds a broadcast tree. As has been previously discussed, the cost of building and maintaining a broadcast tree is power expensive. Rather than maintaining a broadcast tree, the proposed protocol regularly synchronizes to obtain local information that the protocol uses in making data communication decisions. These decisions do not depend on the decisions of other nodes, so expensive tree maintenance is avoided.

Wieselthier assumes that a LMH can transmit at any power level from 0 to infinity. While unlimited transmission power may be helpful when building a minimum power transmission tree, it is not a reasonable assumption. In the proposed protocol, the initial transmission power of each node is known and fixed.

Finally, Wieselthier's protocol deals only with data broadcast. There is no peer messaging and no data query. In fact, the size and contents of the database and data broadcast are not specified.

With these factors in mind, the proposed protocol compares favorably with the Wieselthier protocol and is an improvement by providing more realistic assumptions, greater detail and more methods for data communication.

7.3 Conclusions for Proposed MANET Data Communication Protocol

The purpose of this research was to develop a MANET data communication protocol that provided all three forms of MANET data communication while conserving the power of the battery operated LMHs and SMHs. As part of the research a MANET data communication benchmark was developed to allow more even evaluation of this protocol and comparison of this protocol to other MANET data communication protocols.

The proposed protocol, TriM, compared favorably to existing MANET data communication protocols while providing the additional capability of peer-to-peer messaging.

There are large differences in the performance of TriM between the military and rescue scenarios on one side and the business scenario on the other. This is due to the environment rather than the protocol. The protocol did not perform data query and peer messaging well when the network was sparsely populated. However, it is difficult to imagine how any protocol would provide these services more capably in a network as sparse as the military and rescue scenarios. Additional work is necessary in sparsely populated regions to determine appropriate ways to provide data communication services.

In the more heavily populated business scenario, data query and peer messaging were more successful and TriM performed well. Considering the scenarios suggested in the literature, a MANET is either sparsely populated or heavily populated. This needs further investigation.

One important aspect of a benchmark is that it collaborative and widely accepted within the appropriate research community. This benchmark serves as a starting point to that collaboration. Further work in publicizing the benchmark and refining it through collaboration is an area for future research.

Using the benchmark, analysis and simulation, a MANET data communication protocol was proposed. The proposed protocol was successful in a number of ways. First, it provided all three forms of MANET data communication in an orderly and power aware manner. In addition, the protocol compared favorably to other protocols previously proposed in the research literature.

At the completion of this research, a MANET data communication benchmark had been developed and proposed and a power-aware MANET data communication protocol capable of all three forms of MANET data communication was developed, analyzed and simulated. This was the purpose of the research undertaken, and this purpose was achieved.

7.4 MANET Data Communication Protocol Research Future Work

While conducting this work, ideas for improvement and further research were noted. It seems clear that additional foundational work is necessary before significant advances in MANET data communication can take place. Several of these will be advanced.

The development of the MANET scenarios requires additional work. This work lies in the further characterization of the scenarios. One important piece is the development of appropriate mobility models. When the movement characteristics of nodes within the scenario are better parameterized, it may be found that current protocols work extremely well or extremely poorly. For instance, a random distribution of nodes on the battlefield is probably not the best mobility model. However, no mobility model that only models individual movement is appropriate. Soldiers in battle travel in squads and squads move in a coordinated fashion. While a battlefield MANET appears sparsely populated, it may in fact be well populated in the area of the region where the nodes are moving in a coordinated fashion. Other potential scenarios should also be pursued and evaluated as potential MANET applications.

In addition to mobility models, the effect of increased transmission ranges should be investigated. In this research, an unstated assumption was that the Hamming distance was 1. The ability to increase coverage through data relay and greater node cooperation should be studied. The cost to the nodes relaying data must be carefully considered in terms of time and power consumption.

The benchmark proposed in this research needs collaborative effort with others working in this area. This work would be to continue the definition and validation of this and other MANET benchmarks. As no other MANET benchmarks currently exist, this is an area that may provide an extensive amount of research potential.

Further work on the protocol itself is in order. Adding real-time capabilities, directional antennas, variable power transmissions, etc. provide an ongoing list of items that can be added to a new or modified protocol. In addition, some design decisions can be changed. For instance, determining when is a broadcast reasonable is a question that needs further investigation.

BIBLIOGRAPHY

[1] ACM SIGMOBILE Preliminary Program. Association of Computing Machinery. 9 Dec. 2003 <http://sigmobile.org/mobicom/2003/program.html>.

[2] ANSI/IEEE Std. 802.11, 1999 Edition. 1999.

[2b] Bluetooth. 9 Dec. 2003 <http://www.bluetooth.com>.

[3] Dallas Convention Center Floor Plans. Dallas Convention Center. 9 Dec. 2003 <http://www.dallasconventioncenter.com/facility/floor.asp>.

[4] Fairfax County Fire and Rescue. 9 Dec. 2003 <http://www.vatf1.org/index.shtml>.

[5] FEMA. Urban Search and Rescue (US&R) Incident Response Team (IRT) Operation Manual. 9356.2-PR), 2000.

[6] MANET Charter. IETF MANET Working Group. 9 Dec. 2003 <http://www.ietf.org/html.charters/manet-charter.html>.

[7] Section 2.5.3. In *Proc. 54th Internet Engineering Task Force* July, 2002.

[8] Odeum Sports and Expo Center Floor Plan, DuPage County, Illinois. 9 Dec. 2003 <http://www.odeumexpo.com/floor.htm>.

[9] Sacramento California Urban Search and Rescue. 9 Dec. 2003 <http://www.cityofsacramento.org/fire/usar.html>.

[10] Standard Performance Evaluation Corporation. 9 Dec. 2003 <http://www.spec.org>.

[11] The Computer Vision Homepage. Carnegie Mellon University. 9 Dec. 2003 <http://www-2.cs.cmu.edu/~cil/vision.html>.

[12] The Internet Engineering Task Force (IETF). 9 Dec. 2003 <http://www.ietf.org/>.

[13] Transaction Processing Performance Council. 9 Dec 2003 <http://www.tpc.org>.

[14] Acharya, S., Franklin, M., and Zdonik, S. Balancing Push and Pull for Data Broadcast. In *Proc. 1997 ACM SIGMOD International Conference on Management of Data (SIGMOD '97)*, pp. 183-194, Tucson, Arizona, USA, May, 1997.

[15] Acharya, S. and Muthukrishnan, S. Scheduling On-demand Broadcasts: New Metrics and Algorithms. In *Proc. Fourth Annual International Conference on Mobile Computing and Networking (MOBICOM '98)*, pp. 43-54, Dallas, Texas, USA, October, 1998.

[16] Aggelou, G. and Tafazalli, R. RDMAR: A Bandwidth-Efficient Routing Protocol for Mobile Ad Hoc Networking. In *Proc. Second International Workshop on Wireless Mobile Multimedia (WoWMoM)*, pp. 26-33, Seattle, Washington, USA, August, 1999.

[17] Aksoy, D. and Franklin, M. Scheduling for Large-Scale On-Demand Data Broadcasting. In *Proc. 17th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '98)*, pp. 651-659, San Francisco, California, USA, March-April, 1998.

[18] Banik, S., Energy-Efficient Transaction Management for Real-Time Mobile Databases in Ad-Hoc Network Environments. Thesis, University of Oklahoma, 2001.

[19] Basagni, S., Chlamtac, I., Syrotiuk, V., and Woodward, B. A Distance Routing Algorithm for Mobility (DREAM). In *Proc. Fourth International Conference on Mobile Computing and Networking (MOBICOM '98)*, pp. 76-84, Dallas, Texas, USA, October, 1998.

[20] Bitton, D., Dewitt, D., and Turbyfill, C. Benchmarking Database Systems - A Systematic Approach. In *Proc. Ninth International Conference on Very Large Databases (VLDB 1983)*, pp. 8-19, Florence, Italy, October-November, 1983.

[21] Brian, M., "How Microprocessors Work," *Howstuffworks, Inc.* 2002.

[22] Broch, J., Maltz, D., Johnson, D., Hu, Y., and Jetcheva, J. A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols. In *Proc. Fourth International Conference on Mobile Computing and Networking (MOBICOM '98)*, pp. 85-97, Dallas, Texas, USA, October, 1998.

[23] Bull, J., Smith, L., Pottage, L., and Freeman, R. Benchmarking Java against C and Fortran for Scientific Applications. In *Proc. ACM Java Grande - International Symposium on Computing in Object-oriented Parallel Environments (Java Grande/ISCOPE 01)*, pp. 97-105, Stanford, California, USA, June, 2001.

[24] Carey, M., Dewitt, D., and Naughton, J. The OO7 Benchmark. In *Proc. 1993 ACM SIGMOD International Conference on Management of Data (SIGMOD '93)*, pp. 12-21, Washington, D.C., USA, May, 1993.

[25] Celik, A. and Datta, A. A Scalable Approach for Broadcasting Data in a Wireless Network. In *Proc. Fourth ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, pp. 131-138, Rome, Italy, July, 2001.

[26] Chin, K. and Kumar, M., "AMTree: An Active Approach to Multicasting in Mobile Networks," *Mobile Networks and Applications*, 6 pp. 361-376, 2001.

[27] Chung, Y. and Kim, M. QEM: A Scheduling Method for Wireless Broadcast Data. In *Proc. Sixth International Conference on Database Systems for Advanced Applications (DASFAA '99)*, pp. 135-142, Hsinchu, Taiwan, April, 1999.

[28] Corson, M., Freebergyser, J., and Sastry, A., "Mobile Ad Hoc Networking: Editorial," *Mobile Networks and Applications*, 4(3): pp. 137-138, 1999.

[29] Darmont, J., Petit, B., and Schneider, M. OCB: A Generic Benchmark to Evaluate the Performances of Object-Oriented Database Systems. In *Proc. Advances in Database Technology - LNCS 1377 (EDBT '98)*, pp. 326-340, Valencia Spain, March, 1998.

[30] Das, S., Castañeda, R., and Yan, J., "Simulation-Based Performance Evaluation of Routing Protocols for Mobile Ad Hoc Networks," *Mobile Networks and Applications*, 5(3): pp. 179-189, 2000.

[31] Datta, A., VanderMeer, D., Kim, J., Celik, A., and Kumar, V. Adaptive Broadcast Protocols to Support Efficient and Energy Conserving Retrieval from Databases in Mobile Computing Environments. A TimeCenter Technical Report (University of Arizona, 1977.

[32] DiMicco, J. Mobile Ad Hoc Voting. In *Proc. Workshop on Mobile Ad Hoc Collaboration* April, 2002.

[33] Duhl, J. and Damon, C. A Performance Comparison of Object and Relational Databases Using the Sun Benchmark. In *Proc. Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA '88)*, pp. 153-163, San Diego, California, USA, September, 1998.

[34] Durst, R., Scott, K., Zukoski, M., and Raghavendra, C. Metrics for Comparing Multicast Protocols in Mobile Ad Hoc Networks. In *Proc. Aerospace Conference* March, 2001.

[35] Fife, L. and Gruenwald, L., "Research Issues for Data Communication in Mobile Ad-Hoc Network Database Systems," *ACM SIGMOD Record*, 32(2): June 2003.

[36] Grassi, V. Prefetching Policies for Energy Saving and Latency Reduction in a Wireless Broadcast Delivery System. In *Proc. 3rd ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, pp. 77-84, Boston, Massachusetts, USA, August, 2000.

[37] Gruenwald, L., Javed, M., and Gu, M. Energy-Efficient Data Broadcasting in Mobile Ad-Hoc Networks. In *Proc. International Database Engineering and Applications Symposium (IDEAS '02)*, July, 2002.

[38] Guo, Y., Pinotti, M., and Das, S., "A New Hybrid Broadcast Scheduling Algorithm for Assymetric Communication Systems," *ACM Mobile Computing and Communications Review*, 5(3): pp. 39-54, July 2001.

[39] Haas, Z., Pearlman, M., and Samar, P. The Zone Routing Protocol (ZRP) for Ad Hoc Networks. Jul. 2002. IETF, 1 Aug 2002 <http://www.ietf.org/internet-drafts/draft-ietf-manet-zone-zrp-02.txt>.

[40] Hammond, S. and Loft, R. Architecture and Application: The Performance of the NEC SX-4 on the NCAR Benchmark Suite. In *Proc. Supercomputing '96 (SC '96)*, Pittsburg, Pennsylvania, USA, November, 1996.

[41] Imielinski, I., Viswanathan, S., and Badrinath, B. Energy Efficient Indexing on Air. In *Proc. ACM SIGMOD International Conference on Management of Data (SIGMOD 94)*, pp. 25-36, May, 1994.

[42] Imielinski, I., Viswanathan, S., and Badrinath, B., "Data on Air: Organization and Access," *IEEE Transactions on Knowledge and Data Engineering*, 9(3): pp. 353-372, May/June 2002.

[43] Johansson, P., Larsson, T., Hedman, N., Mielczarek, B., and Degermark, M. Scenario-Based Performance Analysis of Routing Protocols for Mobile Ad-Hoc Networks. In *Proc. 5th International Conference on Mobile Computing and Networking (MOBICOM '99)*, pp. 195-206, August, 1999.

[44] Johnson, D., Maltz, D., Hu, Y., and Jetcheva, J. The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks. Feb. 2002. IETF. 1 Aug 2002. <http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-07.txt>.

[45] Jones, C., Sivalingam, K., Agrawal, P., and Chen, J., "A Survey of Energy Efficient Protocols for Wireless Networks," *Wireless Networks*, 7 pp. 343-358, 2001.

[46] Jung, I., You, Y., Lee, J., and Kim, K. Broadcasting and Caching Policies for Location-Dependent Queries in Urban Environments. In *Proc. Second International Symposium on Mobile Commerce (WMC '02)*, pp. 54-60, Atlanta, Georgia, USA, September, 2002.

[47] Keum, C., Choi, W., Hong, E., Kim, W., and Whang, K. Performance Evaluation of Replica Control Algorithms in a Locally Distributed Database System. In *Proc. 4th International Conference on Database Systems for Advanced Applications (DASFAA 95)*, pp. 388-396, April, 1995.

[48] Ko, Y. and Vaidya, N. Location-Aided Routing (LAR) in Mobile Ad Hoc Networks. In *Proc. Fourth International Conference on Mobile Computing and Networking (MOBICOM '98)*, pp. 66-75, Dallas, Texas, USA, October, 1998.

[49] Kravets, R. and Krishnan, P. Power Management Techniques for Mobile Communication. In *Proc. Fourth International Conference on Mobile Computing and Networking (MOBICOM '98)*, pp. 157-168, Dallas, Texas, USA, October, 1998.

[50] Kumar, S., Pires, L., Ponnuswamy, S., Nanavati, C., Golusky, J., Vojta, M., Wadi, S., Pandalai, D., and Spaanenburg, H. A Benchmark Suite for Evaluating Configurable Computing Systems - Status, Reflections, and Future Directions. In *Proc. Eighth ACM International Symposium on Field-Programmable Gate Arrays (FPGA 2000)*, pp. 126-134, Monterey, California, USA, February, 2000.

[51] Kunz, T. and Cheng, E. On-Demand Multicasting in Ad-Hoc Networks: Comparing AODV and ODMRP. In *Proc. 22nd International Conference on Distributed Computing Systems (ICDCS 2002)*, Vienna, Austria, July, 2002.

[52] Lam, K., Chan, E., and Yuen, J. Broadcast Strategies to Maintain Cached Data for Mobile Computing Systems. In *Proc. Advances in Database Technologies - LNCS 1552 (ER '98)*, pp. 193-204, November, 1998.

[52b] Lau, C. and Gruenwald, L. Managing Time-Critical Transactions in Mobile Ad-Hoc Network Databases. Manuscript, 2003.

[53] Lee, S., Su, W., and Gerla, M., "Wireless Ad Hoc Multicast Routing with Mobility Prediction," *Mobile Networks and Applications*, 6(4): pp. 351-360, 2001.

[54] Leong, H. and Si, A. Data Broadcasting Strategies over Multiple Unreliable Wireless Channels. In *Proc. Fourth International Conference on Information and Knowledge Management* pp. 96-104, November-December, 1995.

[55] Lettieri, P., Fragouli, C., and Srivastava, M. Low Power Error Control for Wireless Links. In *Proc. 3rd International Conference on Mobile Computing and Networking (MOBICOM '97)*, pp. 139-150, September, 1997.

[56] Li, Q., Aslam, J., and Rus, D. Online Power-aware Routing in Wireless Ad-hoc Networks. *(SIGMOBILE '01)*, pp. 97-107, Rome, Italy, July, 2001.

[57] Lin, X., GPS Based Localized Routing Algorithms for Wireless Networks. Master's Thesis, Computer Science, University of Ottawa, 2000.

[58] Liu, J., Zhang, Q., Li, B., Zhu, W., and Zhang, J., "A Unified Framework for Resource Discovery and QoS-Aware Provider Selection in Ad Hoc Networks," *ACM Mobile Computing and Communications Review*, 6(1): pp. 13-21, 2002.

[59] Müller, H., Müller, W., Marchand-Maillet, S., Pun, T., and Squire, D., "A Framework for Benchmarking in CBIR," *Multimedia Tools and Applications*, 21 pp. 55-73, 2003.

[60] Nasipuri, A. and Castañeda, R., "Performance of Multipath Routing for On-Demand Protocols in Mobile Ad Hoc Networks," *Mobile Networks and Applications*, 6(4): pp. 339-349, 2001.

[61] Obraczka, K. and Viswanath, K., "Flooding for Reliable Broadcast in Multi-Hop Ad Hoc Networks," *Wireless Networks*, 7 pp. 627-634, 2001.

[61b] O'Reilly, J. and Lilegdon, W. Introduction to AweSim. In *Proc. 1999 Winter Simulation Conference*, pp. 196-200, 1999.

[62] Perkins, C., Belding-Royer, E., and Das, S. Ad Hoc On Demand Distance Vector (AODV) Routing. Jun. 2002. IETF. 1 Aug 2002 <http://www.ietf.org/internet-drafts/draft-ietf-manet-aodv-11.txt>.

[62b] Pritsker, A. and O'Reilly, J. *Simulation with Visual SLAM and AweSim,* 2nd ed. New York: John Wiley & Sons, 1999.

[63] Roos, T., Myllymäki, P., and Tirri, H., "A Statistical Modeling Approach to Location Estimation in Mobile Ad Hoc Networks," *IEEE Transactions on Mobile Computing*, 1(1): pp. 59-69, 2002.

[64] Sass, P., "Communication Networks for the Force XXI Digitized Battlefield," *Mobile Networks and Applications*, 4 pp. 139-155, 1999.

[65] Seydim, A. and Dunham, M. A Location Dependent Benchmark with Mobility Behavior. In *Proc. International Database Engineering and Applications Symposium (IDEAS 2002)*, pp. 74-85, Edmonton, Canada, July, 2002.

[66] Sim, S., Easterbrook, S., and Holt, R. Using Benchmarking to Advance Research: A Challenge to Software Engineering. In *Proc. 25th International Conference on Software Engineering (ICSE 2003)*, pp. 74-83, Portland, Oregon, USA, May, 2003.

[67] Singh, S., Woo, M., and Raghavendra, C. Power Aware Routing in Mobile Ad Hoc Networks. In *Proc. 4th International Conference on Mobile Computing and Networking (MOBICOM '98)*, pp. 181-190, October, 1998.

[68] Tang, C., Raghavendra, A., and Prasanna, V. Energy Efficient Adaptation of Multicast Protocols in Power Controlled Wireless Ad Hoc Networks. In *Proc. International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN02)*, Manila, Phillipines, May, 2002.

[69] Tseng, Y., Ni, S., Chen, Y., and Shue, J., "The Broadcast Storm Problem in a Mobile Ad Hoc Network," *Wireless Networks*, 8 pp. 153-167, 2002.

[70] Wieselthier, J., Nguyen, G., and Ephremides, A. On the Construction of Energy-Efficient Broadcast and Multicast Trees in Wireless Networks. In *Proc. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2000)*, pp. 585-594, Tel Aviv, Israel, March, 2000.

[71] Wieselthier, J., Nguyen, G., and Ephremides, A., "Algorithms for Energy-Efficient Multicasting in Static Ad Hoc Wireless Networks," *Mobile Networks and Applications*, 6(4): pp. 251-263, 2001.

[72] Wieselthier, J., Nguyen, G., and Ephremides, A. Resource-Limited Energy-Efficient Wireless Multicast of Session Traffic. In *Proc. 34th Hawaii International Conference on System Sciences (HICSS-34)*, January, 2001.

[73] Williams, B. and Camp, T. Comparison of Broadcasting Techniques for Mobile Ad Hoc Networks. In *Proc. International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC 2002)*, pp. 194-205, June, 2002.

[74] Wu, S., Ni, S., Tseng, Y., and Shen, J. Route Maintenance in Wireless Ad Hoc Network. In *Proc. 33rd Hawaii International Conference on System Sciences (HICSS-33)*, Wailea, Maui, Hawaii, USA, January, 2000.

[75] Xu, Y., Heidemann, J., and Estrin, D. Geography-Informed Energy Conservation for Ad Hoc Routing. In *Proc. Seventh International Conference on Mobile Computing and Networking (MOBICOM 2001)*, pp. 70-84, Rome, Italy, July, 2001.

[76] Xuan, P., Sen, S., González, O., Fernandez, J., and Ramamritham, K. Broadcast on Demand: Efficient and Timely Dissemination of Data in Mobile Environments. In *Proc. Third IEEE Real-Time Technology and Applications Symposium (RTAS '97)*, pp. 38-48, Montreal, Canada, June, 1997.

[77] Yajima, E., Hara, T., Tsukamoto, M., and Nishio, S. Scheduling and Caching Strategies for Broadcasting Correlated Data. In *Proc. 16th ACM Symposium on Applied Computing (SAC 2001)*, pp. 504-509, Las Vegas, Nevada, USA, March, 2001.

[78] Zorzi, M. and Naghshineh, M., "Conference Report on the Seventh Annual ACM International Conference on Mobile Computing and Networking (Mobicom 2001)," *ACM Mobile Computing and Communications Review*, 5(2): pp. 2-4, July 2001.

Appendix A

Benchmark and Protocol Parameters

A.1 Protocol Parameters

In Chapter 3, a MANET data communication protocol is proposed. Table A-1 contains all variables associated with the protocol with a brief description.

Parameter	Description
bcastLen	Length: Broadcast Period
bcastPrep	Length: Broadcast Preparation
$density_{SMH}$	Average number of SMHs reachable by each LMHs transmission.
endSC	End of Idle Stage & Service Cycle
idleLen	Length: Idle Stage
$items_{dyn}$	Maximum number of items in the dynamic portion of the broadcast.
$items_{stat}$	Number of items in the static portion of the broadcast.
numLMH	Number of LMHs
pullLen	Length: Data Pull Stage
pushLen	Length: Data Push Stage: bcastPrep + bcastLen
reqFreq	Average number of requests made by a SMH.
serve	Maximum time to serve one data request.
startSC	Start of the Service Cycle
startIdle	End of Data Pull Stage / Start of Idle Stage
startPull	End of Data Push Stage / Start of Data Pull Stage
startPush	End of Synchronization Stage / Start of Data Push Stage
synchLen	Length: Synchronization Stage: syncLMH + syncSMH
synchLMH	Length: LMH Synchronization
synchSMH	Length: SMH Synchronization
$transmit_{data}$	Time to transmit each data item in the broadcast.
$transmit_{idx}$	Time to transmit each entry in the broadcast index.
$transmit_{LMH}$	Time for one LMH to transmit its unique ID and location
$transmit_{SMH}$	Time for one SMH to transmit its unique ID and location

Table A-1 – Protocol Parameters

A.1 Benchmark Parameters

In Chapter 4, a MANET data communication benchmark was proposed. Table A-2 gives all variables associated with the benchmark evaluation parameters with a brief description and the location used.

Parameter	Description	Used in Equations
avgBcastEff	Average broadcast effectiveness for all broadcasts	4-10
avgPwrL	Average LMH power dissipation per unit time	4-7
avgPwrS	Average SMH power dissipation per unit time	4-6
$bcastEff_m$	Broadcast effectiveness of broadcast m	4-9, 4-10
$intItems_k$	Number of items in broadcast of interest to SMH k	4-9
$itemsBCast_k$	Number of items broadcast to SMH k	4-9
$msgRec_k$	Number of peer messages received by SMH k	4-12
numBcast	Number of broadcasts	4-8, 4-10
$numHeard_b$	The number of SMHs detecting a LMH during synchronization period b	4-8
numLMH	Number of LMHs deployed in system	4-7
numSMH	Number of SMHs deployed in system	4-6, 4-8, 4-9, 4-11, 4-12
peerEfficiency	Ratio of peer messages arriving to messages sent	4-12
peerFreq	The rate of peer messages per SMH during one S. C.	4-12
perCvr	The average percentage of SMHs hearing a data broadcast	4-8
period	Length of simulation period in seconds	4-6, 4-7
queryEfficiency	Ratio of data queries served to data queries sent	4-11
rec_k	Time each node k spends in Receive Mode	4-6, 4-7
recRateL	LMH power dissipation rate in transmit mode	4-7
recRateS	SMH power dissipation rate in transmit mode	4-6
reqFreq	The rate of data queries per SMHs in one S.C.	4-11
$stby_k$	Time each node k spends in Standby Mode	4-6, 4-7
stbyRateL	LMH power dissipation rate in transmit mode	4-7
stbyRateS	SMH power dissipation rate in transmit mode	4-6
$totalServed_k$	The number of data queries served for SMH k	4-11
$trans_k$	Time each node k spends in transmit mode	4-6, 4-7
transRateL	LMH power dissipation rate in transmit mode	4-7
transRateS	SMH power dissipation rate in transmit mode	4-6

Table A-2 – Benchmark Parameters

A.3 Evaluation Parameters

In Chapter 5, the proposed MANET data communication protocol is evaluated. The variables associated with this evaluation, a brief description and the location of each is shown in Tables A-3, A-4 and A-5. Table A-3 shows the evaluation parameters for the power consumption of LMHs. These are calculated in Equations 5-5 to 5-8.

Parameter	Description	Used in Equations
avgPwrL	Average power consumed by a LMH	5-5
avgRec	Average time LMH spends in receive mode	5-5, 5-7
avgStby	Average time LMH spends in standby mode	5-5, 5-8
avgTrans	Average time LMHs spends in transmit mode	5-5, 5-6
bcastLen	Length of broadcast transmission period	5-8
bcastPrep	Length of broadcast preparation period	5-7
idleLen	Length of idle stage	5-8
$items_{dyn}$	Number of dynamic items in broadcast	5-6
$items_{stat}$	Number of static items in broadcast	5-6
peerRec	number of routing requests LMH receives	5-6
period	Length of simulation	5-5
probL	Probability only LMHs heard during synchronization	5-6, 5-7, 5-8
probLS	Probability LMHs and SMHs heard during synch	5-6, 5-7, 5-8
probS	Probability only SMHs heard during synchronization	5-6, 5-7, 5-8
pullLen	Length of data pull stage	5-6
qryRec	Number of data queries LMH receives	5-6
recRateL	Power dissipation of LMH in receive mode	5-5
stbyRateL	Power dissipation of LMH is standby mode	5-5
synchLen	Length of synchronization stage	5-7
$transmit_{data}$	Time for LMH to transmit one data item	5-6
$transmit_{idx}$	Time for LMH to transmit one index item	5-6
$transmit_{LMH}$	Time for LMH to transmit SMH synch data	5-5, 5-6, 5-7
transRateL	Power dissipation of LMH is transmit mode	5-5
$transmit_{rte}$	Time for LMH to route one peer message	5-6

Table A-3 – Evaluation Parameters – LMH Power Consumption

Table A-4 provides the evaluation parameters for SMH power consumption. These are covered in Equations 5-1 to 5-4.

Parameter	Description	Used in Equations
avgPwrS	Average power consumed by a SMH	5-1
avgRec	Average time SMH spends in receive mode	5-1, 5-3
avgStby	Average time SMH spends in standby mode	5-1, 5-4
avgTrans	Average time SMHs spends in transmit mode	5-1, 5-2
bcastLen	Length of broadcast transmission period	5-4
bcastPrep	Length of broadcast preparation period	5-4
idleLen	Length of idle stage	5-4
$items_{dyn}$	Number of dynamic items in broadcast	5-3
$items_{stat}$	Number of static items in broadcast	5-3
numHeard	Number of LMH heard by SMH in synchronization	5-3
peerFreq	Frequency of peer messages	5-2
period	Length of simulation	5-1
probB	Probability of SMH hearing a LMH broadcast	5-3, 5-4
probInt	Probability a dynamic item is of interest to a SMH	5-3
probL	Probability only LMHs heard during synchronization	5-2, 5-3
probLS	Probability LMHs and SMHs heard during synch	5-2, 5-3
probS	Probability only SMHs heard during synchronization	5-2, 5-3
pullLen	Length of data pull stage	5-2, 5-3
recRateS	Power dissipation of SMH in receive mode	5-1
reqFreq	Frequency of data queries	5-2
stbyRateS	Power dissipation of SMH is standby mode	5-1
synchLen	Length of synchronization stage	5-3
$transmit_{data}$	Time for LMH to transmit one data item	5-3
$transmit_{idx}$	Time for LMH to transmit one index item	5-3
$transmit_{msg}$	Time for SMH to transmit one peer message	5-2
$transmit_{qry}$	Time for SMH to transmit one data query	5-2
$transmit_{SMH}$	Time for LMH to transmit SMH synch data	5-2, 5-3
transRateS	Power dissipation of SMH is transmit mode	5-1

Table A-4 – Evaluation Parameters – SMH Power Consumption

Table A-5 shows the remaining evaluation parameter variables from Chapter 5.

Parameter	**Description**	**Used in Equations**
$items_{dyn}$	Number of dynamic items in broadcast	5-10
$items_{stat}$	Number of static items in broadcast	5-10
localSMH	Number of SMHs querying a specific LMH	5-11
numHeard	Number of LMH heard by SMH in synchronization	5-10
numLMH	Number of LMHs in the network	5-9
peerFreq	Average number of non-routed peer messages	5-12
peerMsg	Average number of peer messages sent	5-12
peerRte	Average number of peer messages routed	5-12
probInt	Probability a dynamic item is of interest to a SMH	5-10
probL	Probability only LMHs heard during synchronization	5-12
probLS	Probability LMHs and SMHs heard during synch	5-12
probR	Probability a SMH can reach a LMH	5-11
probS	Probability only SMHs heard during synchronization	5-12
regionHeight	Height of simulation region	5-9
regionWidth	Width of simulation region	5-9
reqFreq	Number of data queries made by a SMH	5-11
transArea	Transmission area of one LMH	5-9

Table A-5 – Other Evaluation Parameters

Appendix B

AweSim Simulation C Insert Code

The AweSim simulation tool allows the user to define a network through discrete events graphically. This network then shows the flow of the network from one discrete event to the next. In Figure B-1, the network is shown graphically.

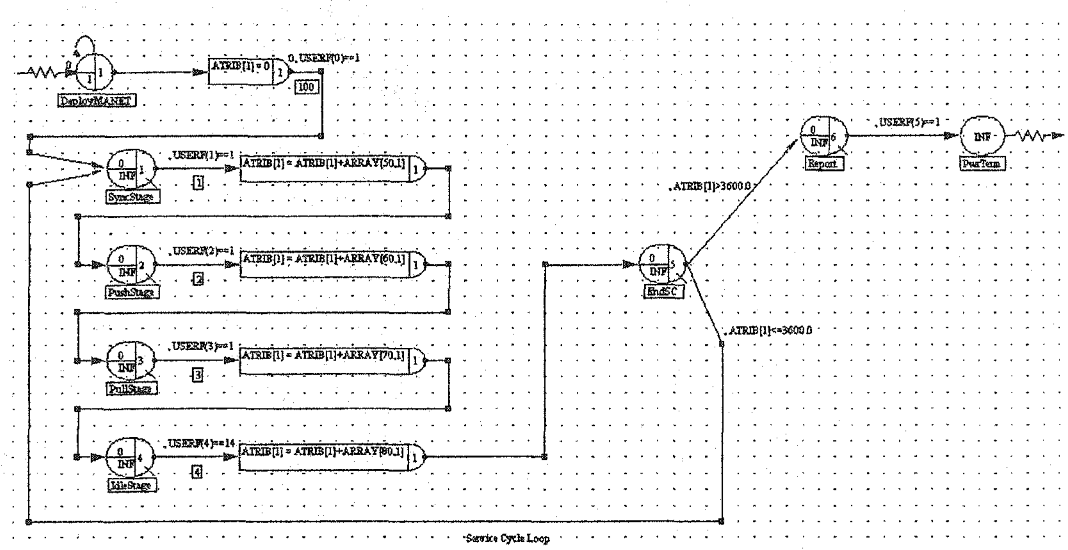


Figure B-1 AweSim Simulation Model for MANET Protocol

As part of the simulation tool, the user is able to write code inserts that define the behavior at each discrete event. This allows the user to set simulation parameters, track behavior, and record the results of the simulation. The user inserts for the MANET data communication protocol described in this research were written using Microsoft Visual C, one of the few choices provided by the tool.

In the following pages are the code inserts provided to define the behavior of the MANET data communication protocol simulation. The code for only one scenario is shown. The scenario provided is for a military scenario with a small database, small broadcast, and a low level of peer messages and data queries. There are few differences between this and other scenarios. The primary differences are in the number of LMHs and SMHs, the size of the region, and the initial placement of nodes in the simulation region. In the military scenario, LMHs and SMHs are initialized in a small region, a so-called battle line. They move randomly once the simulation starts throughout the region. In the domestic rescue and temporary business network scenarios, the nodes are initialized at random locations throughout the simulation region.

```
// 08 November 2003
//
// Leslie D. Fife
// Dissertation MANET Simulations
//
// SMALL DBASE
// SMALL BCAST
// SMALL Queries
// SMALL Msgs
//
// MILITARY
//
#include "vslam.h"
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>
//
//#define Scenario 1       // Business Scenario
//#define Scenario 2       // Rescue Scenario
#define Scenario 3         // Military Scenarion
//*******************************************************************
//     Scenario 3 - MILITARY NODES start in a small region together, and
//     then move throughout the battle. They all start with the same x
//     value (x=0) and are scattered over a 1 km line (1000 meters).
//*******************************************************************
//
// Network Parameters Military Scenario
//
#define NumLMH              20            // Number of LMH
#define NumSMH              1000          // Number of SMH
//
//*******************************************************************
//                                        // Power Dissipation (watts/hr)
#define ActRateL            170.0         // LMH Active
#define DzRateL             20.0          // LMH Doze
#define SlpRateL            2.0           // LMH Sleep
#define ActRateS            7.0           // SMH Active
#define DzRateS             1.0           // SMH Doze
#define SlpRateS            0.1           // SMH Sleep
//
//*******************************************************************
//
#define BandL       2000         // Bandwidth LMH (kbps)
#define BandS       100          // Bandwidth SMH (kbps)
#define RadiusL     250          // Transmission Radius LMH (m)
#define RadiusS     100          // Transmission Radius SMH (m)
#define CPUL        1700         // CPU Power LMH (MIPS)
#define CPUS        100          // CPU Power SMH (MIPS)
#define PwrL        1            // Battery Power LMH (watts)
#define PwrS        1            // Battery Power SMH (watts)
#define MinMob      0            // Minimum Mobility (m/sec)
#define MaxMob      20           // Maximum Mobility (m/sec)
#define Period      3600.0       // Length of simulation (sec)
#define RegionX     10000        // Region Width (m)
#define RegionY     15000        // Region Height (m)
//
// Transmission Times / Item Sizes
//
#define IdxSize     128          // Size of one index item. (byte)
```

```
#define IdxTrans    0.000064     // Transmit Time Index Item
                                 //   by LMH (sec)
#define DataSize    64           // Size of data item in broadcast.
                                 //   (Kbyte)
#define DataTrans   0.032        // Time to transmit Broadcast Data
                                 //   Item by LMH (sec)
#define QrySize     256          // Size of SMH data query (byte)
#define QryTrans    0.00256      // Time to transmit data query from
                                 //   SMH (sec)
#define MsgSize     512          // Size of peer message. (byte)
#define MsgTrans    0.00512      // Time to transmit peer message
                                 //   by SMH (sec)
#define RteTrans    0.000256     // Time to route peer message by
                                 //   LMH (sec)
#define SynchSzL    5            // Size of synch message LMH (bytes)
#define SynchTransL 0.0000025    // Time to transmit synch message
                                 //   for 1 LMH (sec)
#define SynchSzS    6            // Size of synch message SMH (bytes)
#define SynchTransS 0.00006      // Time to transmit synch message
                                 //   for SMH (sec)
//
// Variable Workload Parameters
//
#define DBaseSz     500          // Items in database
//#define DBaseSzS  500          // Items in small database
//#define DBaseSzM  2000         // Items in medium database
//#define DBaseSzL  5000         // Items in large database
#define BCastSz     50           // Items in broadcast 1/2 static
//#define BCastSzS  50           // Items in small broadcast
//#define BCastSzM  100          // Items in medium broadcast
//#define BCastSzL  200          // Items in large broadcast
#define ReqFreq     5            // # data queries per sec.
//#define ReqFreqS  5            // Small # data queries per sec.
//#define ReqFreqM  20           // Medium # data queries per sec.
//#define ReqFreqL  40           // Large # data queries per sec.
#define PeerFreq    5            // # peer messages per sec.
//#define PeerFreqS 5            // Small # peer messages per sec.
//#define PeerFreqM 20           // Medium # peer messages per sec.
//#define PeerFreqL 40           // Large # peer messages per sec.
//
//Stage Parameters
//
#define instrIdx    5
#define routeRate   2            //divisor
//Simulation Constants Parameters
//
const int Lbase = 100;           // Beginning Array Location LMHs
const int Sbase = 200;           // Beginning Array Location SMHs
char * header1 = "Military Scenario";
char * header2 = "Small DBase - Small BCast";
char * header3 = "Small Query - Small Peer Msg";

//*********************************************************************
// xpos()  Randomly assign x position.
//*********************************************************************
int xpos()
{
      switch (Scenario)
      {
        case 1:
```

```
        case 2:
             return rand()%RegionX;
             break;
        case 3:
             return 0; // Battle Line
             break;
      }
}
//*********************************************************************
// ypos()  Randomly assign y position.
//*********************************************************************
int ypos()
{
      switch (Scenario)
      {
        case 1:
        case 2:
             return rand()%RegionY;
             break;
        case 3:
             return rand()%1000; // Battle Line Arrayed
             break;
      }
}
//*********************************************************************
// init_node()  Initializes each SMH and LMH for MANET deployment
//*********************************************************************
int init_node()
{
      int j;
      // Initialize LMHs. Room for up to 100.
      for(j=1; j<= NumLMH; j++)
      {
            PUTARY(Lbase+j,1,xpos());  // x position
            PUTARY(Lbase+j,2,ypos());  // y position
            PUTARY(Lbase+j,3,PwrL);    // power level
            PUTARY(Lbase+j,4,0);       // total time active (sec)
            PUTARY(Lbase+j,5,0);       // total time doze (sec)
            PUTARY(Lbase+j,6,0);       // total time sleep (sec)
            PUTARY(Lbase+j,7,0);       // num broadcasts
            PUTARY(Lbase+j,8,1);       // currently L active?
                                       //   1 - yes 0 - no.
            PUTARY(Lbase+j,9,0);       // # Dynamic Items
            PUTARY(Lbase+j,10,1);      // currently S active?
      }
      // Initialize SMHs. Room for up to 1000.
      for(j=1; j<= NumSMH; j++)
      {
            PUTARY(Sbase+j,1,xpos());  // x position
            PUTARY(Sbase+j,2,ypos());  // y position
            PUTARY(Sbase+j,3,PwrS);    // power level
            PUTARY(Sbase+j,4,0);       // total time active (sec)
            PUTARY(Sbase+j,5,0);       // total time doze (sec)
            PUTARY(Sbase+j,6,0);       // total time sleep (sec)
            PUTARY(Sbase+j,7,0);       // queries sent
            PUTARY(Sbase+j,8,0);       // queries served
            PUTARY(Sbase+j,9,0);       // messages sent
            PUTARY(Sbase+j,10,0);      // messages received
            PUTARY(Sbase+j,11,1);      // currently L active?
                                       //   1 - yes 0 - no.
```

```
            PUTARY(Sbase+j,12,0);       // nearest LMH
            PUTARY(Sbase+j,13,0);       // bcast Sent
            PUTARY(Sbase+j,14,0);       // bcast Interest
            PUTARY(Sbase+j,15,1);       // currently S active?
        }
        return 1;
}
//**********************************************************************
// update(int)  Update sim time
//**********************************************************************
int update(int stage)
{
        double currTime;     // Current Time
        double stageTime;
        switch(stage)
        {
          case 1: // Synch
               stageTime = (double)GETARY(50,1);
               currTime = (double)GETARY(1,1);
               currTime += stageTime;
               PUTARY(1,1,currTime);
               break;
          case 2: // Push
               stageTime = (double)GETARY(60,1);
               currTime = (double)GETARY(1,1);
               currTime += stageTime;
               PUTARY(1,1,currTime);
               break;
          case 3: // Pull
               stageTime = (double)GETARY(70,1);
               currTime = (double)GETARY(1,1);
               currTime += stageTime;
               PUTARY(1,1,currTime);
               break;
          case 4: // Idle
               stageTime = (double)GETARY(80,1);
               currTime = (double)GETARY(1,1);
               currTime += stageTime;
               PUTARY(1,1,currTime);
               break;
        }
        return 1;
}
//**********************************************************************
// init_stages()  Initializes protocol stage parameters.
//**********************************************************************
int init_stages()
{
        double simTime = 0.0;
        double synchL,
               synchS,
               synchLen;            // Length of LMH and SMH portions
                                    //   of the Synch Stage
        double bcastPrep,
               bcastLen,
               pushLen;
        double pullLen;
        double idleLen;             // Length of Idle Stage
        double scTime;
        //
```

```
        // Sim Time
        //
        PUTARY(1,1,simTime);
        //
        // Synch Node Parameters
        //
        synchL = NumLMH * SynchTransL;
        synchS = (NumSMH / NumLMH) * SynchTransS;
        synchLen = synchL+synchS;
        PUTARY(50,1, synchLen);
        PUTARY(50,2, synchL);
        PUTARY(50,3, synchS);
        //
        // Push Node Parameters
        //
        bcastPrep = (instrIdx*BCastSz)/(CPUL*1000000.0);
        bcastLen = ((BCastSz*IdxTrans)+(BCastSz*DataTrans))*NumLMH;
        pushLen = bcastPrep + bcastLen;
        PUTARY(60,1,pushLen);
        PUTARY(60,2,bcastPrep);
        PUTARY(60,3,bcastLen);
        PUTARY(61,1,0); // BCast Eff - Totaled
        PUTARY(61,2,0); // Num Bcasts
        PUTARY(61,3,0); // Num Missed Total
        PUTARY(61,4,0); // Num Missed SC
        //
        // Pull Node Parameters
        //
        pullLen = (ReqFreq*DataTrans)*(NumSMH/NumLMH);
        PUTARY(70,1,pullLen);
        //
        // Idle Node Parameters
        //
        // Initially set to 0, to compare to other protocols.
        // Later adjusted to show the effect of this stage.
        //
        idleLen = 0;
        PUTARY(80,1,idleLen);
        scTime=synchLen+pushLen+pullLen+idleLen;
        PUTARY(40,1,scTime);
        return 1;
}
//***********************************************************************
// init_net()  controls the initialization of a new MANET deployment.
//***********************************************************************
int init_net()
{
        init_node();          // Initial LMHs and SMHs.
        init_stages();        // Initalize Protocol Stage Parameters
        return 1;
}
//***********************************************************************
// dist(int x1, y1, x2, y2)       Calculate the distance (in meters)
//                                between 2 points. (Euclidean Distance)
//***********************************************************************
double dist(int x1, int y1, int x2, int y2)
{
        return sqrt((x1-x2)*(x1-x2) + (y1-y2)*(y1-y2));
}
//***********************************************************************
```

```
// reach()  Calculates and stores hearable and reachable nodes for each
//          LMH and SMH.
//          Sets SMHs Inactive if no LMH heard.
//***********************************************************************
int reach()
{
	int i, j;
	int x1, x2;
	int y1, y2;
	double tmpD;
	int actFlag = 0;
	FILE * id1;		// SMH hear/reach SMH
	FILE * id2;		// SMH reach LMH
	FILE * id3;		// SMH hear LMH
	FILE * id4;		// LMH hear/reach LMH
	FILE * id5;		// LMH hear/reach SMH
	id1 = fopen("smhsmh.txt", "w");
	id2 = fopen("smhRlmh.txt", "w");
	id3 = fopen("smhHlmh.txt", "w");
	id4 = fopen("lmhlmh.txt", "w");
	id5 = fopen("lmhsmh.txt", "w");
	fprintf(id1, "SMH SMH\n");
	fprintf(id2, "SMH Reaches LMH\n");
	fprintf(id3, "SMH Hears LMH\n");
	fprintf(id4, "LMH LMH\n");
	fprintf(id5, "LMH SMH\n");
	for (i=1; i<=NumSMH; i++)  // SMH SMH
	{
		x1 = (int)GETARY(Sbase+i,1);		// x position
		y1 = (int)GETARY(Sbase+i,2);		// y position
		fprintf(id1, "%d", i);
		for (j=1; j<=NumSMH; j++)
		{
			x2 = (int)GETARY(Sbase+j,1);		// x position
			y2 = (int)GETARY(Sbase+j,2);		// y position
			tmpD = dist(x1,y1,x2,y2);
			if( tmpD < RadiusS && i !=j)
				fprintf(id1, " %d %.2f", j, tmpD);
		}
		fprintf(id1, "\n");
	}
	for (i=1; i<=NumSMH; i++)  // SMH R LMH & SMH H LMH
	{
		x1 = (int)GETARY(Sbase+i,1);		// x position
		y1 = (int)GETARY(Sbase+i,2);		// y position
		fprintf(id2, "%d", i);
		fprintf(id3, "%d", i);
		actFlag = 0;
		for (j=1; j<=NumLMH; j++)
		{
			x2 = (int)GETARY(Lbase+j,1);		// x position
			y2 = (int)GETARY(Lbase+j,2);		// y position
			tmpD = dist(x1,y1,x2,y2);
			if(tmpD < RadiusS)
				fprintf(id2, " %d %.2f", j, tmpD);
			if(tmpD < RadiusL)
			{
				fprintf(id3, " %d %.2f", j, tmpD);
				actFlag++;
			}
```

```
        }
        PUTARY(Sbase+i, 11, actFlag); // Set SMH as BCAST
                                        // active.
                                        // 0 if no LMHs heard
                                        // # heard otherwise.
        fprintf(id2, "\n");
        fprintf(id3, "\n");
    }
    for (i=1; i<=NumLMH; i++)  // LMH LMH
    {
        x1 = (int)GETARY(Lbase+i,1);      // x position
        y1 = (int)GETARY(Lbase+i,2);      // y position
        fprintf(id4, "%d", i);
        for (j=1; j<=NumLMH; j++)
        {
            x2 = (int)GETARY(Lbase+j,1);      // x position
            y2 = (int)GETARY(Lbase+j,2);      // y position
            tmpD = dist(x1,y1,x2,y2);
            if(tmpD < RadiusL && i !=j)
                fprintf(id4, " %d %.2f", j, tmpD);
        }
        fprintf(id4, "\n");
    }
    for (i=1; i<=NumLMH; i++)  // LMH SMH
    {
        x1 = (int)GETARY(Lbase+i,1);      // x position
        y1 = (int)GETARY(Lbase+i,2);      // y position
        fprintf(id5, "%d", i);
        actFlag = 0;
        for (j=1; j<=NumSMH; j++)
        {
            x2 = (int)GETARY(Sbase+j,1);      // x position
            y2 = (int)GETARY(Sbase+j,2);      // y position
            tmpD = dist(x1,y1,x2,y2);
            if(tmpD < RadiusS)
            {
                fprintf(id5, " %d %.2f", j, tmpD);
                actFlag++;
            }
        }
        PUTARY(Lbase+i, 8, actFlag); // Set LMH as BCAST
                                        // active.
                                        // 0 if no SMHs heard
                                        // # heard otherwise.
        fprintf(id5, "\n");
    }
    fclose(id1);
    fclose(id2);
    fclose(id3);
    fclose(id4);
    fclose(id5);
    return 1;
}
//*********************************************************************
// addSynchTime()  Add time in Stby to ALL nodes (SMH and LMH)
//*********************************************************************
int addSynchTime()
{
    double transTimeS, transTimeL;
    double recTimeS, recTimeL;
```

```
        double synchTime;
        double tmpTime;
        int j;
        synchTime = (double)GETARY(50,1);
        transTimeS = SynchTransS;
        transTimeL = SynchTransL;
        recTimeS = synchTime - transTimeS;
        recTimeL = synchTime - transTimeL;
        // LMHs.
        //
        for(j=1; j<= NumLMH; j++)
        {
                tmpTime = (double)GETARY(Lbase+j,4);  // Time in Trans
                tmpTime += transTimeL;
                PUTARY(Lbase+j,4,tmpTime);          // new time
                tmpTime = (double)GETARY(Lbase+j,5);  // Time in Rec
                tmpTime += recTimeL;
                PUTARY(Lbase+j,5,tmpTime);          // new time
        }
        // SMHs.
        //
        for(j=1; j<= NumSMH; j++)
        {
                tmpTime = (double)GETARY(Sbase+j,4);  // Time in Trans
                tmpTime += transTimeS;
                PUTARY(Sbase+j,4,tmpTime);          // new time
                tmpTime = (double)GETARY(Sbase+j,5);  // Time in Rec
                tmpTime += recTimeS;
                PUTARY(Sbase+j,5,tmpTime);          // new time
        }
        return 1;
}
//*********************************************************************
// synch_node()       controls the behavior of each node as they perform
//                    the synchronization stage of the Tri-Modal
//                    MANET data communication protocol.
//*********************************************************************
int synch_node (ENTITY peCur)
{
        reach();  //Calculate which nodes can hear and reach other nodes
        addSynchTime();
        update(1);
        return 1;
}
//*********************************************************************
// addPushTime()  Add time in Push to ALL nodes (SMH and LMH)
//      To do this, the size of the broadcast/index of each LMH must
//      be calculated and the number of dynamic items hearable by
//      each SMH must be calculated for each LMH and SMH not deactivated
//      during synchronizaiton for not hearing other nodes.
//*********************************************************************
int addPushTime()
{
        double prepTime, prepLMH;
        double bcastTime, bcastLMH, bcastSMH;
        double pushTime;
        double tmpTime, tmpNum;
        int numDyn, numStat, numBcast, numMiss;
        int i,j;
        char junk[81];
```

```
int j1, j2;
int heard[NumSMH+1]; // #dynamic heard
int intItems[NumSMH+1], bcastItems[NumSMH+1], flag;
FILE * data;
FILE * test;
data = fopen("smhHlmh.txt", "r");
test = fopen("test.txt", "w");
pushTime = (double)GETARY(60,1);
prepTime = (double)GETARY(60,2);
bcastTime = (double)GETARY(60,3);
// LMHs.
//
for(j=1; j<= NumLMH; j++)
{
        // Broadcast Prep - Either Rec or Stby.
        //
        if((int)GETARY(Lbase+j,8)==0)
                prepLMH=0.0;
        else
        {
                numBcast = (int)GETARY(Lbase+j,7);
                numBcast++;
                PUTARY(Lbase+j,7,numBcast);
                numStat = BCastSz/2;
                numDyn = (int)GETARY(Lbase+j,9);
                prepLMH=(numStat+numDyn)*instrIdx/(CPUL*1000000.0);
        }
        tmpTime = (double)GETARY(Lbase+j,5);  // Time in Rec
        tmpTime += prepLMH;
        PUTARY(Lbase+j,5,tmpTime);            // new time
        tmpTime = (double)GETARY(Lbase+j,6);  // Time in Stby
        tmpTime += (prepTime - prepLMH);
        PUTARY(Lbase+j,6,tmpTime);            // new time
        //
        // Broadcast - Transmit or Stby.
        //
        if((int)GETARY(Lbase+j,8)==0)
                bcastLMH=0.0;
        else
        {
                //numStat = BCastSz/2;                    // Calc above
                //numDyn = (int)GETARY(Lbase+j,9); // Calc above
                bcastLMH=(numStat+numDyn)*(IdxTrans+DataTrans);
        }
        tmpTime = (double)GETARY(Lbase+j,4);  // Time in Trans
        tmpTime += bcastLMH;
        PUTARY(Lbase+j,4,tmpTime);            // new time
        tmpTime = (double)GETARY(Lbase+j,6);  // Time in Stby
        tmpTime += (bcastTime - bcastLMH);
        PUTARY(Lbase+j,6,tmpTime);            // new time
}
// SMHs.
//
fgets(junk, 80, data);
for(i=1;i<=NumSMH;i++)
{
  fscanf(data,"%d", &j1);
  heard[i] = 0;
  intItems[i] = 0;
  bcastItems[i] = 0;
```

```
        flag = 0;
        for(j=1; j<=(int)GETARY(Sbase+i,11); j++)
        {
             fscanf(data,"%d%d", &j1, &j2);
             heard[i] += (int)GETARY(Lbase+j1,9);
             if(flag==0)
               intItems[i] += numStat;
             intItems[i] += heard[i];
             bcastItems[i] += numStat;
             bcastItems[i] += heard[i];
             flag++;
        }
        PUTARY(Sbase+i,13,bcastItems[i]);
        PUTARY(Sbase+i,14,intItems[i]);
        fprintf(test,"%d %d\n",i, heard[i]);
      }
      numMiss = 0;
      for(j=1; j<= NumSMH; j++)
      {
             tmpTime = (double)GETARY(Sbase+j,6);  // Time in Stby
             tmpTime += prepTime;
             PUTARY(Sbase+j,6,tmpTime);          // new time
             if((int)GETARY(Sbase+j,11)==0)
             {
                    bcastSMH = 0.0;
                    numMiss++;
             }
             else
             {
                    bcastSMH=(numStat+heard[j])*(IdxTrans+DataTrans);
             }
             tmpTime = (double)GETARY(Sbase+j,5);  // Time in Rec
             tmpTime += bcastSMH;
             PUTARY(Sbase+j,5,tmpTime);          // new time
             tmpTime = (double)GETARY(Sbase+j,6);  // Time in Stby
             tmpTime += (bcastTime - bcastSMH);
             PUTARY(Sbase+j,6,tmpTime);          // new time
      }
      PUTARY(61,4,numMiss);                 // # SMH miss Bcast SC
      tmpNum = (int)GETARY(61,3);
      tmpNum += numMiss;
      PUTARY(61,3,tmpNum);                  // # SMH miss Bcast total
      tmpNum = (int)GETARY(61,2);
      tmpNum++;
      PUTARY(61,2,tmpNum);                  // #Bcast Cycles
      fclose(data);
      fclose(test);
      return 1;
}
//***********************************************************************
// reportMid()      BCast Eff,
//***********************************************************************
int reportMid(void)
{
      int j;
      double bcastEff = 0.0,
             totInt = 0.0,
             totBcast = 0.0;
      double tmpNum,
             perMiss = 0.0,
```

```
            totMiss = 0.0;
        FILE * rep;
        rep=fopen("report.txt","a");
        // Calculate BCast Effectiveness
        //
        totBcast = 0.0;
        totInt = 0.0;
        for(j=1; j<= NumSMH; j++)
        {
              tmpNum = (double)GETARY(Sbase+j,13);
              totBcast += tmpNum;
              tmpNum = (double)GETARY(Sbase+j,14);
              totInt += tmpNum;
        }
        if ((int)totBcast == 0)
              bcastEff = 0.0;
        else
              bcastEff = (totInt / totBcast)*100.0;
        tmpNum = (double)GETARY(61,1);
        tmpNum += bcastEff;
        PUTARY(61,1,tmpNum);
        tmpNum = (double)GETARY(61,2);
        fprintf(rep,"BCast  %3.0f  Effectiveness  (percent)=  %8.2f\n",tmpNum,
        bcastEff);
        totMiss = (double)GETARY(61,4);
        perMiss = totMiss / NumSMH;
        perMiss *= 100.0;
        fclose(rep);
        return 1;
}
//*************************************************************************
// push_node()       controls the behavior of each node as they perform
//                   the Data Push stage of the Tri-Modal MANET data
//                   communication protocol. This is the time for data
//                   broadcast.
//*************************************************************************
int push_node (ENTITY peCur)
{
        addPushTime();
        update(2);
        reportMid();
        return 1;
}
//*************************************************************************
// activePull()  Calculates and stores hearable and reachable nodes for
//            each LMH and SMH. Sets SMHs Inactive if no LMH heard.
//*************************************************************************
int activePull()
{
        int i, j;
        int x1, x2;
        int y1, y2;
        double tmpD, minD, minLMH=0;
        int actFlag = 0;
        //
        // LMH Active?
        // 0 - No
        // 1 - Yes
        //
        for (i=1; i<=NumLMH; i++)  // LMH LMH
```

```
{
        x1 = (int)GETARY(Lbase+i,1);        // x position
        y1 = (int)GETARY(Lbase+i,2);        // y position
        actFlag = 0;
        for (j=1; j<=NumLMH; j++)
        {
                x2 = (int)GETARY(Lbase+j,1);        // x position
                y2 = (int)GETARY(Lbase+j,2);        // y position
                tmpD = dist(x1,y1,x2,y2);
                if(tmpD < RadiusL && i!=j)
                        actFlag++;
        }
        if(actFlag > 0)
                PUTARY(Lbase+i, 8, 1); // Set LMH L active
        else
                PUTARY(Lbase+i, 8, 0); // Set LMH L inactive
        actFlag = 0;
        for (j=1; j<=NumSMH; j++)
        {
                x2 = (int)GETARY(Sbase+j,1);        // x position
                y2 = (int)GETARY(Sbase+j,2);        // y position
                tmpD = dist(x1,y1,x2,y2);
                if(tmpD < RadiusS)
                        actFlag++;
        }
        if(actFlag > 0)
                PUTARY(Lbase+i, 10, 1); // Set LMH S active
        else
                PUTARY(Lbase+i, 10, 0); // Set LMH S inactive
}
//
// SMH Active?
// 0 - No
// 1 - Yes
//
for (i=1; i<=NumSMH; i++)  // SMH SMH
{
        x1 = (int)GETARY(Sbase+i,1);        // x position
        y1 = (int)GETARY(Sbase+i,2);        // y position
        actFlag = 0;
        for (j=1; j<=NumSMH; j++)
        {
                x2 = (int)GETARY(Sbase+j,1);        // x position
                y2 = (int)GETARY(Sbase+j,2);        // y position
                tmpD = dist(x1,y1,x2,y2);
                if( tmpD < RadiusS && i !=j)
                        actFlag++;
        }
        if(actFlag>0)
                PUTARY(Sbase+i, 15, 1); // Set SMH S active
        else
                PUTARY(Sbase+i, 15, 0); // Set SMH S inactive
        actFlag = 0;
        minLMH = 0;
        for (j=1; j<=NumLMH; j++)
        {
                x2 = (int)GETARY(Lbase+j,1);        // x position
                y2 = (int)GETARY(Lbase+j,2);        // y position
                tmpD = dist(x1,y1,x2,y2);
                if(tmpD < RadiusS)
```

```
                {
                        actFlag++;
                        if(minLMH==0)
                        {
                                minD = tmpD;
                                minLMH = j;
                        }
                        else
                        if(tmpD < minD)
                        {
                                minD = tmpD;
                                minLMH = j;
                        }
                }
        }
        if(actFlag>0)
                PUTARY(Sbase+i, 11, 1); // Set SMH L active
        else
                PUTARY(Sbase+i, 11, 0); // Set SMH L inactive
        PUTARY(Sbase+i, 12, minLMH); // Set closest LMH
    }
    return 1;
}
//*******************************************************************
// addPullTime()  Add time in Pull to ALL nodes (SMH and LMH)
//      To do this, the workload at each active node must
//      be calculated and the number of items served, messages sent
//      and routing requests handled must be calculated. Finally,
//      the number of unserved data queries must be stored for use
//      in the next SC data push stage.
//*******************************************************************
int addPullTime()
{
    int qLoad[NumLMH+1];        // # queries at each LMH
    int qNum[NumLMH+1]; // # SMH querying each LMH
    int qNot[NumLMH+1]; // # queries not served (for SC Bcast)
    int qNotAvg[NumLMH+1];
    int mLoad[NumLMH+1];
    int activeS,
        activeL,         // Active?
        actSMH;                 // # SMHs active
    int nearLMH;
    int i, j, numSrv, Srvd,
        qSent, qRec, mSent, mRec, tmp;
    double pullTime, currTime, tmpTime;
    pullTime = (double)GETARY(70,1);
    for(j=1; j<=NumLMH; j++)
    {
        qLoad[j]=0;
        qNum[j]=0;
        qNot[j]=0;
        qNotAvg[j]=0;
        mLoad[j]=0;
    }
    actSMH=0;
    // Create query loads
    for(i=1; i<=NumSMH; i++)
    {
        activeS = (int)GETARY(Sbase+i,11);
        if(activeS != 0)
```

```
		{
			nearLMH = (int)GETARY(Sbase+i,12);
			qLoad[nearLMH] += ReqFreq;
			qNum[nearLMH]++;
			actSMH++;
		}
	}
	// Create msg loads
	for(i=1; i<=NumLMH; i++)
	{
		mLoad[i]=(qNum[i]*PeerFreq)/routeRate;
	}
	// Calculate queries served and time LMH in each power mode.
	for(i=1; i<=NumLMH; i++)
	{
		activeL = (int)GETARY(Lbase+i,8);
		activeS = (int)GETARY(Lbase+i,10);
		if((activeS != 0)&&(activeL != 0)) // Can Rte/Serve
		{
			currTime=pullTime;
			currTime -= mLoad[i]*RteTrans;
			numSrv = (int)(currTime/QryTrans);
			if(numSrv > qLoad[i])
			{
				currTime -= qLoad[i]*QryTrans;
				qNot[i] = 0;
				Srvd=qLoad[i];
			}
			else
			{
				currTime -= numSrv*QryTrans;
				qNot[i]=qLoad[i]-numSrv;
				Srvd=numSrv;
				qNotAvg[i]=qNot[i]/qNum[i];
				if(qNot[i]>(BCastSz/2))
					qNot[i]=BCastSz/2;
			}
			tmpTime = (double)GETARY(Lbase+i,4);   // Trans
			tmpTime += (pullTime-currTime);
			PUTARY(Lbase+i,4,tmpTime);            // new time
			tmpTime = (double)GETARY(Lbase+i,5);   // Time in Rec
			tmpTime += currTime;
			PUTARY(Lbase+i,5,tmpTime);            // new time
		}
		else
		if(activeL != 0) // Can Route
		{
			currTime = pullTime;
			currTime -= mLoad[i]*RteTrans;
			qNot[i] = 0;
			tmpTime = (double)GETARY(Lbase+i,4);   // Trans
			tmpTime += (pullTime-currTime);
			PUTARY(Lbase+i,4,tmpTime);            // new time
			tmpTime = (double)GETARY(Lbase+i,5);   // Time in Rec
			tmpTime += currTime;
			PUTARY(Lbase+i,5,tmpTime);            // new time
		}
		else
		if(activeS != 0) // Can Serve
		{
```

```
            currTime=pullTime;
            numSrv = (int)(currTime/QryTrans);
            if(numSrv > qLoad[i])
            {
                  currTime -= qLoad[i]*QryTrans;
                  qNot[i] = 0;
                  Srvd=qLoad[i];
            }
            else
            {
                  currTime -= numSrv*QryTrans;
                  qNot[i]=qLoad[i]-numSrv;
                  Srvd=numSrv;
                  qNotAvg[i]=qNot[i]/qNum[i];
                  if(qNot[i]>(BCastSz/2))
                        qNot[i]=BCastSz/2;
            }
            tmpTime = (double)GETARY(Lbase+i,4);  // Trans
            tmpTime += (pullTime-currTime);
            PUTARY(Lbase+i,4,tmpTime);          // new time
            tmpTime = (double)GETARY(Lbase+i,5);  // Time in Rec
            tmpTime += currTime;
            PUTARY(Lbase+i,5,tmpTime);          // new time
      }
      else // Stby entire time.
      {
            qNot[i] = 0;
            tmpTime = (double)GETARY(Lbase+i,6);  // Stby
            tmpTime += pullTime;
            PUTARY(Lbase+i,6,tmpTime);          // new time
      }
      PUTARY(Lbase+i,9,qNot[i]);                // # dynamic
}
// Calculate queries sent/served and msgs sent/received
// and time SMH in each power mode.
//
for(i=1; i<=NumSMH; i++)
{
      mSent = PeerFreq;
      qSent = ReqFreq;
      mRec = 0;
      qRec = 0;
      activeS = (int)GETARY(Sbase+i,15);
      activeL = (int)GETARY(Sbase+i,11);

      if((activeS != 0)&&(activeL != 0)) // Trans and Rec Modes
      {
            mRec=PeerFreq;
            nearLMH = (int)GETARY(Sbase+i,12);
            if(qNot[nearLMH]==0)
                  qRec=ReqFreq;
            else
                  qRec=qNotAvg[nearLMH];
            //
            currTime = pullTime;
            currTime -= mSent*MsgTrans;
            currTime -= qSent*QryTrans;
            if(currTime < 0) currTime = 0;
            tmpTime = (double)GETARY(Sbase+i,4);  // Trans
            tmpTime += (pullTime-currTime);
```

```
        PUTARY(Sbase+i,4,tmpTime);          // new time
        tmpTime = (double)GETARY(Sbase+i,5);  // Time in Rec
        tmpTime += currTime;
        PUTARY(Sbase+i,5,tmpTime);          // new time
}
else
if(activeL != 0)
// NO SMHs near
// NO non routed peer
{
        mRec=PeerFreq/2;
        nearLMH = (int)GETARY(Sbase+i,12);
        if(qNot[nearLMH]==0)
                qRec=ReqFreq;
        else
                qRec=qNotAvg[nearLMH];
        //
        currTime = pullTime;
        currTime -= (mSent/2.0)*MsgTrans;
        currTime -= qSent*QryTrans;
        if(currTime < 0) currTime = 0;
        tmpTime = (double)GETARY(Sbase+i,4);  // Trans
        tmpTime += (pullTime-currTime);
        PUTARY(Sbase+i,4,tmpTime);          // new time
        tmpTime = (double)GETARY(Sbase+i,5);  // Time in Rec
        tmpTime += currTime;
        PUTARY(Sbase+i,5,tmpTime);          // new time
}
else
if(activeS != 0) // No Rte Peer
// NO LMHs near
// NO routed peer, no query
{
        mRec=PeerFreq/2;
        qRec=0;
        //
        currTime = pullTime;
        currTime -= (mSent/2.0)*MsgTrans;
        tmpTime = (double)GETARY(Sbase+i,4);  // Trans
        tmpTime += (pullTime-currTime);
        PUTARY(Sbase+i,4,tmpTime);          // new time
        tmpTime = (double)GETARY(Sbase+i,5);  // Time in Rec
        tmpTime += currTime;
        PUTARY(Sbase+i,5,tmpTime);          // new time
}
else // Stby entire time.
{
        mRec=0;
        qRec=0;
        tmpTime = (double)GETARY(Sbase+i,6);  // Stby
        tmpTime += pullTime;
        PUTARY(Sbase+i,6,tmpTime);          // new time
}
tmp = (int)GETARY(Sbase+i,7);
tmp += qSent;
PUTARY(Sbase+i,7,tmp);
tmp = (int)GETARY(Sbase+i,8);
tmp += qRec;
PUTARY(Sbase+i,8,tmp);
tmp = (int)GETARY(Sbase+i,9);
```

```
            tmp += mSent;
            PUTARY(Sbase+i,9,tmp);
            tmp = (int)GETARY(Sbase+i,10);
            tmp += mRec;
            PUTARY(Sbase+i,10,tmp);
        }
        return 1;
}
//*********************************************************************
// pull_node()        controls the behavior of each node as they perform
//                    the Data Pull stage of the Tri-Modal MANET data
//                    communication protocol. This stage includes
//                    data-query and peer-2-peer communication.
//*********************************************************************
int pull_node (ENTITY peCur)
{
        activePull();
        addPullTime();
        update(3);
        return 1;
}
//*********************************************************************
// newxpos()  Calculates new x pos based on mobility and direction.
// newypos()  Calculates new y pos based on mobility and direction.
//*********************************************************************
int newxpos(int x)
{
        int xdist;
        int xdir;
        double elapseTime;
        elapseTime = (double)GETARY(40,1);
        xdist = (int)((rand()%MaxMob)*elapseTime);
        xdir = rand()%2;
        if(xdir==0)
              xdist *= -1;
        x += xdist;
        if(x < 0) x = 0;
        if(x > RegionX) x = RegionX;
        return x;
}
int newypos(int y)
{
        int ydist;
        int ydir;
        double elapseTime;
        elapseTime = (double)GETARY(40,1);
        ydist = (int)((rand()%MaxMob)*elapseTime);
        ydir = rand()%2;
        if(ydir==0)
              ydist *= -1;
        y += ydist;
        if(y < 0) y = 0;
        if(y > RegionY) y = RegionY;
        return y;
}
//*********************************************************************
// move()  Move all nodes (SMH and LMH)
//*********************************************************************
int move()
{
```

```
	int j;
	int x, y;
	// Move LMHs.
	for(j=1; j<= NumLMH; j++)
	{
		x = (int)GETARY(Lbase+j,1);	// x position
		y = (int)GETARY(Lbase+j,2);	// y position
		x = newxpos(x);
		y = newypos(y);
		PUTARY(Lbase+j,1,x);		// new x position
		PUTARY(Lbase+j,2,y);		// new y position
	}
	// Move SMHs.
	for(j=1; j<= NumSMH; j++)
	{
		x = (int)GETARY(Sbase+j,1);	// x position
		y = (int)GETARY(Sbase+j,2);	// y position
		x = newxpos(x);
		y = newypos(y);
		PUTARY(Sbase+j,1,x);		// new x position
		PUTARY(Sbase+j,2,y);		// new y position
	}
	return 1;
}
//**********************************************************************
// addIdleStby()  Add time in Stby to ALL nodes (SMH and LMH)
//**********************************************************************
int addIdleStby()
{
	double idleTime;
	double stbyTime;
	int j;
	// Move LMHs.
	idleTime = (double)GETARY(80,1);
	for(j=1; j<= NumLMH; j++)
	{
		stbyTime = (double)GETARY(Lbase+j,6); // Time in Stby
		stbyTime += idleTime;
		PUTARY(Lbase+j,6,stbyTime);		// new Stby time
	}
	// SMHs.
	for(j=1; j<= NumSMH; j++)
	{
		stbyTime = (double)GETARY(Sbase+j,6); // Time in Stby
		stbyTime += idleTime;
		PUTARY(Sbase+j,6,stbyTime);		// new Stby time
	}
	return 1;
}
//**********************************************************************
// idle_node()		controls the behavior of each node as they perform
//			the Idle stage of the Tri-Modal MANET data
//			communication protocol.
//**********************************************************************
int idle_node (void)
{
	move();
	addIdleStby();
	update(4);
	return 1;
```

```
}
//**********************************************************************
// reportStart()     Start of SIM. Print Report Headers
//**********************************************************************
int reportStart(void)
{
	double testsynch, testpush, testpull, testidle, testsc;
	FILE * rep;
	rep=fopen("report.txt","w");
	fprintf(rep, "%s\n", header1);
	fprintf(rep, "%s\n", header2);
	fprintf(rep, "%s\n\n", header3);
	testsynch=(double)GETARY(50,1);
	testpush=(double)GETARY(60,1);
	testpull=(double)GETARY(70,1);
	testidle=(double)GETARY(80,1);
	testsc=(double)GETARY(40,1);
	fprintf(rep, "Synch Time %10.5f sec.\n",testsynch);
	fprintf(rep, "Push Time  %10.5f sec.\n",testpush);
	fprintf(rep, "Pull Time  %10.5f sec.\n",testpull);
	fprintf(rep, "Idle Time  %10.5f sec.\n",testidle);
	fprintf(rep, "SC Time    %10.5f sec.\n\n",testsc);
	fprintf(rep,"\n----------------------------------------------\n");
	fflush(rep);
	fclose(rep);
	return 1;
}
//**********************************************************************
// reportEnd()      End of SIM. Print node values.
//**********************************************************************
int reportEnd(void)
{
	int i, j;
	double transSMH = 0.0,
	       recSMH = 0.0,
	       stbySMH = 0.0;
	double transLMH = 0.0,
	       recLMH = 0.0,
	       stbyLMH = 0.0;
	double tmpVal=0.0;
	double avgPwrSMH = 0.0,
	       avgPwrLMH = 0.0;
	double totEff = 0.0,
	       numBC = 0.0,
	       avgBCeff = 0.0;
	double perMiss = 0.0;
	double percentSat = 0.0;
	double reqSat = 0.0,
	       reqMade = 0.0;
	double peerEff = 0.0;
	double msgRec = 0.0,
	       msgSent = 0.0;
	FILE * rep;
	rep=fopen("report.txt","a");
	// Calculate AvgPwr LMH
	//
	for(j=1; j<= NumLMH; j++)
	{
		tmpVal = (double)GETARY(Lbase+j,4);
		transLMH += tmpVal;
```

```
        tmpVal = (double)GETARY(Lbase+j,5);
        recLMH += tmpVal;
        tmpVal = (double)GETARY(Lbase+j,6);
        stbyLMH += tmpVal;
}
avgPwrLMH += transLMH*ActRateL;
avgPwrLMH += recLMH*DzRateL;
avgPwrLMH += stbyLMH*SlpRateL;
avgPwrLMH /= Period;
avgPwrLMH /= NumLMH;
// Calculate AvgPwr SMH
//
for(j=1; j<= NumSMH; j++)
{
        tmpVal = (double)GETARY(Sbase+j,4);
        transSMH += tmpVal;
        tmpVal = (double)GETARY(Sbase+j,5);
        recSMH += tmpVal;
        tmpVal = (double)GETARY(Sbase+j,6);
        stbySMH += tmpVal;
        tmpVal = (double)GETARY(Sbase+j,7);
        reqMade += tmpVal;
        tmpVal = (double)GETARY(Sbase+j,8);
        reqSat += tmpVal;
        tmpVal = (double)GETARY(Sbase+j,9);
        msgSent += tmpVal;
        tmpVal = (double)GETARY(Sbase+j,10);
        msgRec += tmpVal;
}
avgPwrSMH += transSMH*ActRateS;
avgPwrSMH += recSMH*DzRateS;
avgPwrSMH += stbySMH*SlpRateS;
avgPwrSMH /= Period;
avgPwrSMH /= NumSMH;
percentSat = (reqSat / reqMade)*100.0;
// AvgBcastEff
//
totEff = (double)GETARY(61,1);
numBC =  (double)GETARY(61,2);
avgBCeff = totEff/numBC;
perMiss = (double)GETARY(61,3);
perMiss /= numBC;
perMiss /= NumSMH;
perMiss *= 100;
// AvgResponseTime
//
// Peer Efficiency
//
peerEff = (msgRec / msgSent)*100.0;
// Report Final Stats
fprintf(rep,"\n-------------------------------------------------\n");
fprintf(rep,"Final Simulation Values\n");
fprintf(rep,"=================================================\n");
fprintf(rep,"Avg SMH Power Consumption =      %8.2f watts/hr\n",
avgPwrSMH);
fprintf(rep,"Avg LMH Power Consumption =      %8.2f watts/hr\n",
avgPwrLMH);
fprintf(rep,"Avg pct. SMH Hearing Broadcast = %8.2f\n",
(100.0-perMiss));
fprintf(rep,"Avg Broadcast Effectiveness =    %8.2f\n",
```

```
        avgBCeff);
        fprintf(rep,"Query Efficiency =                 %8.2f\n",
        percentSat);
        fprintf(rep,"Peer Efficiency =                  %8.2f\n",
        peerEff);
        fclose(rep);
        return 1;
}
//*********************************************************************
// USERF()    controls the call to the appropriate stage of the service
//            cycle in the tRI-mODAL MANET data communication protocol.
//*********************************************************************
double USERF(int IFN, ENTITY *peCur)
{
        double simTime;
        FILE * cnt;
        switch(IFN)
        {
                case 0: // Network Initialization
                        init_net();
                        reportStart();
                        break;
                case 1: // Service Cycle
                          // Case 2, 3, 4
                        srand(clock());
                        cnt=fopen("counting.txt", "w");
                        simTime=(double)GETARY(1,1);
                        fprintf(cnt, "Start Time %10.5f\n",simTime);
                        //fflush(cnt);
                        while(simTime <= Period)
                        {
                                synch_node(*peCur);
                                push_node(*peCur);
                                pull_node(*peCur);
                                idle_node();
                                simTime=(double)GETARY(1,1);
                                fprintf(cnt, "Array %10.5f\n",simTime);
                        }
                        fclose(cnt);
                        break;
                case 5: // END Print Report
                        reportEnd();
                        break;
        }
        return 1;
}
```