INTEROPERABILITY OF AD-HOC ROUTING PROTOCOLS

A Thesis by

Madhusudhan Raghuveer

Bachelor of Engineering, Anna University, India, 2005

Submitted to the Department of Electrical Engineering
and the faculty of the Graduate School of
Wichita State University
in partial fulfillment of
the requirements for the degree of
Master of Science

May 2008

INTEROPERABILITY OF AD-HOC ROUTING PROTOCOLS

The following faculty have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science with a major in Electrical Engineering.

____

Ravi Pendse, Committee Chair

____

Kamesh Namuduri, Committee Member

____

Krishna Krishnan, Committee Member

DEDICATION

To my beloved parents and brother

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr.Ravi Pendse, for his continues support, guidance and encouragement throughout the course of my graduate studies at Wichita State University. If it were not for him, I would not have had a chance to prove my worth at Wichita State University, for which I cannot thank him enough. I would like to extend my gratitude to members of my thesis committee,Dr.Kameswara Rao Namuduri and Dr.Krishna Krishnan ,for taking time out to review and evaluate my thesis.

I would like to convey my special thanks to Mr.Nagaraja Thanthry, whose invaluable guidance and suggestions helped me in completing my thesis in a satisfactory and timely manner.

Lastly, I would like to thank my family and friends for there continuous support and encouragement during my course of study in Wichita State University.

ABSTRACT

Mobile ad-hoc networks are fast emerging and play a very important role in the field of wireless communications. With the advantages of mobility and infrastructure less the applications of ad-hoc networks have increased drastically. There are different routing protocols specifically designed for ad-hoc networks. Some examples of these routing protocols are DSDV, DSR AODV and DSR.Each of the routing protocols proposed for ad-hoc networks has its own advantages. For example DSDV is best suited for small scale ad-hoc networks and the design principles are very simple and is not very efficient when there is mobility and the size of the network increases.AODV and DSR are on-demand routing protocols and are best suited for large networks and also when a lot of mobility is involved among the wireless nodes in the network. Thus it is not possible to define a single routing protocol that would best suite all ad-hoc networks. The selection of a routing protocol purely depends on the application and also the scalability of the network.

Having said that there cannot be a single routing protocol for all kinds of ad-hoc network this thesis concentrates on the concept of having interoperability between ad-hoc routing protocols. This thesis proposes a methodology which involves defining a universal packet which would be understood by all routing protocols. The design of having interoperability between the ad-hoc routing protocol involves three phases namely Bootstrap Phase, Route Discovery Phase and Data Exchange Phase. The method involved does not propose any changes to the existing protocols but an addition of universal packet format which would be understood by all routing protocols.

Having discussed the proposed methodology for having interoperability between ad-hoc routing protocols this thesis discuses a mathematical model to calculate the

maximum time involved in the Bootstrap Phase and the Route Discovery phase. The simulation results show that not only the number of hops to the egress node play important role in the time determination in Bootstrap Phase but also the distance between the source and the egress node play a key factor in the time determination process.

TABLE OF CONTENTS

TABLE OF CONTENTS

Chapter Page

LIST OF FIGURES

Chapter 1

Introduction

Wireless technology is one of the fast emerging technologies in the field of networking. The basic design goals are the exchange of information between end users without having any kind of physical connectivity between the devices and also supporting mobility of the nodes. The fast growth in the field of wireless technologies is due to the emergence of devices like laptops, wireless modems and wireless routers that support wireless LAN which give the user the comfort of mobility. As the number of users has increased and mobility was also involved, the design of infrastructure based wireless networks became infeasible. This was the time when the Ad-Hoc network emerged.

Wireless Ad-Hoc Network [1] [10] can be defined as a non infrastructural self configuring collection of wireless nodes. In a wireless Ad-Hoc network there is no concept of having a wireless router for routing the data. Each node plays the role of a mobile router in forming a path to reach from the source to the destination.

The major advantage of Ad-Hoc networks is “On Demand setup” where all the nodes which want to have connectivity form their own network without the dependence on any infrastructure. This becomes a major use in places of disaster recovery, where there needs to be a fast communication set up. There is no need of any infrastructure like mobile routers, as all the nodes can establish connectivity and start the communication process.

The best operation of wireless Ad-Hoc network can be defined when a source wants to communicate with a destination which is not in its radio communication range. Each of the intermediate nodes in the network acts as a router to form a path between the source and the destination to communicate between them. To find the most efficient path between the source and the destination given the various factors like mobility, infrastructure base and packet radio communication features Ad-Hoc Routing Protocols were formulated.

Ad-Hoc Routing Protocols were designed to efficiently handle the basic constraints in wireless communications namely bandwidth, battery power of the nodes and mobility of the nodes. The two major classifications of Ad-Hoc Routing protocols are proactive and reactive routing protocols [2]. In proactive routing protocols, each and every node keeps track of the changes in the network and keeps itself updated with the reachability of a node. Thus, when a node needs to send data to a destination, it knows the reachability of the destination. The major advantage of this is that latency will be reduced when a packet needs to be sent to the destination since the route to the destination is predetermined. The major disadvantage of this protocol is scalability. When the size of the network increases, the number of routing updates increases consuming bandwidth for the route discovery process. Thus, with respect to bandwidth and battery power proactive routing protocols are disadvantageous. Examples of proactive routing protocols are Destination Sequenced Distance Vector Routing (DSDV), Optimized Links State Routing (OLSR) and Wireless Routing Protocol (WRP).

In a Reactive routing Protocol design, the route to a destination is determined only when required .To elaborate, the node reachability information is determined only when there is a need to send data to that destination. Thus when a source node wants to send a packet to a destination, it initiates a Route Request packet and forwards it to its neighbors .The forwarding of the Route Request packet continues until the destination is reached or the reachability of the destination is known to the intermediate nodes. The major advantage of this kind of routing protocol is that it supports large networks, and the overhead on the nodes for maintaining all network information is reduced. The major disadvantage of this kind of routing is that it involves latency when the route needs to be discovered before the forwarding of data can take place. Examples of reactive routing protocols are Ad-Hoc On Demand Distance Vector (AODV) routing and Dynamic Source Routing (DSR).

Thus from the analysis of the two different kinds of Ad-hoc routing protocols, it can be concluded that there is no single Ad-Hoc routing protocol that can be declared the most efficient routing protocol, since each of the routing protocols has its own advantage and disadvantage .The selection of the type of routing protocol totally depends on the network design . The network design is made up of parameters like the size of the network, mobility and various other factors.

Goal of having interoperability between Ad-Hoc Routing Protocols

Having discussed the pros and cons of the different kinds of Ad-Hoc routing protocols, it can be concluded that a single routing protocol cannot be used for all Ad-Hoc routing operations in every domain since each domain might have a set of very important parameters to be satisfied by the Ad-Hoc Routing Protocol being used. For example in a routing domain, mobility of the nodes might be a very important parameter on which the routing protocol selection is dependant whereas mobility of the nodes might be the least priority on routing protocol selection on the other domain. The best thing to do is to have each routing domain be independent on the selection of the routing protocol to be used in their domain but bring up interoperability to have communication between routing domains.

The importance of having interoperability can be best explained with an example by having interoperability between the routing domains playing a major role. An example of Disaster Recovery field where teams from different organizations like military, hospitals and non governmental organizations play a major role in helping the people in the affected areas will be the best scenario to explain the problem. Since these kinds of scenarios happen all of a sudden and it is not possible to design a network in this situation, the best alternative would be to setup a wireless ad-hoc network. In this scenario, when there are three different organizations operating at a given place, each setup their own ad-hoc network and use the routing protocol that will best suit their needs.

The importance of interoperability comes into the picture when organizations like the military and the hospitals would like to communicate with each other, but they use different routing protocols. The other scenario could be when the military would like to communicate with its own team which is being separated by a hospital domain. In both the scenarios, having interoperability between the routing protocols plays a very important role.

Having studied the importance of having interoperability of ad-hoc routing protocols, this research basically concentrates on the implementation of interoperability between ad-hoc routing protocols. The thesis proposes the concept of universal packets which will be understood by all the nodes running different routing protocols and which forms the basis for establishing interoperability of ad-hoc routing protocols. The concept of universal packet format is explained in detail in the upcoming chapters.

The remaining of the thesis is organized as follows. Chapter 2 of the thesis discusses about the literature survey in the field of Adhoc networks. The chapter starts with the background on different types of wireless networks followed by the explanation on the routing principles involved in Adhoc networks. The chapter throws light on the various research work done in the field of Adhoc networks and which also gives an importance of having interoperability in Adhoc Routing Protocols. Chapter 3 explains the design of the new protocol for having interoperability of Adhoc routing protocols. It explains the various phases that would be involved when there needs to be interoperability of Adhoc routing protocols. In chapter4, the author discusses the mathematical model involved in the time calculation for the different phases involved in the protocol design. This is followed by the simulation of the mathematical model to analyze the time taken in each of the phases involved in the design of the protocol. Chapter 5 of the thesis discusses the conclusion of the thesis and the future work involved in this research area.

Chapter 2

Literature Review

This chapter of the thesis will review the various research papers related to this work. Section 2.1 discusses the research papers related to the classification of wireless communication. Section 2.2 briefs the fundamental routing principles based on which all the routing protocols are being built. Section 2.3 explains three major routing protocols in Ad-hoc networks which were built based on the routing principles. Section 2.4 summarizes the findings from the literature survey citing few papers which led to the research work on the interoperability of Ad-hoc Routing Protocols.

2.1 Background

Wireless communications can be broadly divided into two categories.

- Infrastructure Based Networks
- Adhoc Networks

2.1.1 Infrastructure Based Networks:

Infrastructure Based networks [10] can be defined as networks in which there is a have a fixed base station which is responsible for coordinating communication between different wireless nodes. A widely used example of infrastructure based wireless network is Wireless LAN .In Wireless LANs, an access point is used for coordinating the communication between nodes using a particular radio frequency.

2.1.2 Adhoc Networks:

A wireless Ad-Hoc Network [1] can be defined as a network formation between a group of nodes without any pre-established infrastructure. Thus wireless Adhoc Networks can also be defined as infrastructure less networks. In Mobile Adhoc networks each node acts as a mobile router which is capable of establishing a path between a source and destination node and transmitting the data between them in a multihop network scenario.

2.2 Fundamental Principle of Routing:

The basic principle of routing [11] is finding the efficient path to reach the destination from a given source to establish communication between them. The set of rules or the protocol which is used for determining the path to the destination is called a Routing Protocol. Routing Protocols can be broadly classified into the following four categories.

- Distance Vector Routing
- Source Routing
- Link State Routing
- On-Demand Routing

2.2.1 Distance Vector Routing:

Distance Vector Routing Protocol is based on Bellman Ford Algorithm. The basic principle of Distance Vector Routing is that each router transmits its routing table to its neighbor. Each neighbor receiving this routing information builds its routing table based on the information collected. This process of building the routing table is also known as routing by rumor.

2.2.2 Source Routing:

Source Routing is a routing methodology by which the source which is intended to transmit the packet to the destination decides the path through which the packet needs to travel to reach the destination. The intermediate router is not involved in the decision of the next hop router .It just sends the packet to the next hop router as specified by the source router.

2.2.3 Link State Routing:

Link State Routing Protocols [11] are based on the principle that all the routers in the link state routing domain exchange the status of their links to all the other routers in the domain through the use of Link State Advertisement(LSA). With this information being shared, all the routers develop a map of the network and determine the best path to reach any destination. The major advantage of link state routing protocols are its faster convergence time and avoidance of routing loops when compared to distance vector routing. The major disadvantages of this kind of routing algorithm are CPU utilization and memory utilization is high when compared to distance vector routing protocols.

2.2.4 On-Demand Routing Protocols

Unlike source routing, in the case of On-Demand routing [12], the route to a destination is determined only when the source wants to transmit some data. The main advantage of On Demand Routing Protocol when compared to the Source Routing Method is that the source routing addresses the issue of mobility in Adhoc Networks by discovering the route to the destination only when required.

2.3 Routing Protocols in Ad-hoc Networks

The subsequent section is going to discuss the three major routing protocols in Adhoc Networks namely Destination Sequenced Distance Vector Routing, Dynamic Source Routing and Adhoc On-Demand Distance Vector Routing which are based on the routing algorithms discussed in the previous section.

2.3.1 Destination Sequence Distance Vector (DSDV)

The basic principle of DSDV [4] is that each node maintains the shortest path to every destination in the domain along with the neighbor through which the destination can be reached. This principle is based on the concept of Distance Vector Routing in computer networks. At regular interval of time each node broadcasts its route entry to all its neighbors with a sequence number associated with each entry. The purpose of having a sequence number is to have consistency in the routing information exchanged because of the mobility involved in Ad-hoc environment.Whenver there is a change in the Adhoc network the change in the reachability of a particular node is informed to the neighbors and route recalculation is performed, based on the new routing information. The major advantage of DSDV is that since each node has knowledge of how to reach all the nodes in the domain, the packet transmitting time is less as it just needs to transmit the information to the desirable neighbor which can reach the destination. The major disadvantage of this routing methodology will be the route discovery time and convergence. This is because each node needs to know on how to reach the destination**.** Also, the changes that take place in Adhoc environment need to be populated to the entire routing domain. This increases the route discovery overhead in the network.

2.3.2 Dynamic Source Routing (DSR)

In Dynamic Source Routing [5] the source has the complete path information about how to reach a particular destination. In this routing, the basic flow of operations is as follows. Whenever the source wants to send some data to the destination, it checks its routing table and finds the entire path to the destination. The entire path is copied onto the header .When the next hop node receives the packet, it checks if that is the destination. If so, the node accepts the information and processes the packet. Otherwise it forwards the packet to the next node as specified by the source. This process continues until the destination is reached. The three important packets involved in DSR for the establishment of a route are Route Request, Route Reply and Route error. The Route Request packets are generated by the source when they have to discover a route for a particular destination. When the destination receives the Route Request packet it sends a unicast reply with the Route Reply message to the source which originated the route discovery. Route Error packets are basically to inform the source if there is any link breakage along the path from the source to destination. Thus, these are the three packets which are used for route discovery in Dynamic Source Routing. The major advantage of using Dynamic Source Routing is that since the source appends the entire route to the destination with the header after convergence, the exchange of information would be much more efficient when compared to routing protocol based on distance vector routing methods. On the other hand, the disadvantage of DSR would be the convergence time in the route discovery process and the time for hand off when a route request is made for an unreachable destination.

2.3.3Ad-hoc On-Demand Routing (AODV)

The Route discovery process involved in AODV[13] is very similar to that of DSDV which involves Route Request, Route Reply and Route Error as discussed in the previous section. The difference is that the source node does not have the entire route information to the destination as DSR, but it will have only the information of the neighboring node which would transmit the packet to the destination. AODV also maintains a sequence number for each route entry for which it has to check the freshness of the route information and also to avoid looping in the network. The advantage of this routing protocol is that it is suitable for Adhoc Networks which involve a lot of mobility where the route discovery to a destination is initiated only when there is a requirement from the source. The disadvantage of this protocol is the convergence time and the time involved in route discovery process when information needs to be exchanged between the source and the destination.

2.4 Related Work

This section of the thesis is going to bring into picture the various research works that have been done in the field of ad-hoc networks. The study of the work done in the field of ad-hoc network was very useful in the design of the interoperability of adhoc routing protocols .This section gives a brief analysis of different research work that was done and also analyzes why interoperability between ad-hoc routing protocols plays a very important role which is missing in all the papers discussed below.

The RFC 2501 [1] by S.Corson and J.Macker   and the presentation by Chunfeng [10] gives a very good analysis of wireless networks. It is organized in the following manner. First Chunfeng explains of wireless networks, followed by the explanation of the different kinds of wireless networks namely infrastructure based wireless network and Ad-hoc networks. The presentation also gives a very clear picture about the need of Ad-hoc networks with examples and the author also discusses in brief about the major routing protocols involved in wireless ad-hoc network.

"“Routing in ad hoc networks of mobile hosts” by Johnson, D.B [3] explains in detail about the drawbacks of the conventional routing protocol and gives a clear idea of why the conventional routing protocols based on distance vector routing and link state routing do not best suit ad-hoc networks. Having discussed the drawbacks of conventional routing protocols the author also explains the importance of On-Demand routing in ad-hoc networks. To summarize Johnson.D.B explains why conventional routing protocols do not best suit ad-hoc networks and explains the more efficient methodologies that can be adapted for ad-hoc networks.

“Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers” by J.C.Perkins and P.Bhagawat [4] give a very good explanation of the use of Distance Vector routing principles for adhoc networks. The authors also explain the modifications to the Bellman-Ford algorithm that were proposed for the use of distance vector algorithm in a self starting and highly dynamic environment like ad-hoc networks. This paper also adds the modification for the basic network layer routing to support improved MAC layer for the ad-hoc networks.

In the paper "Dynamic Source Routing in Ad Hoc Wireless Networks" by David B. Johnson and David A.Maltz *[5]* the authors give a very good insight on the working of Dynamic Source routing (DSR) .The authors explain the basic operation of Dynamic Source routing protocol followed by how the Route Discovery and Route maintenance happens in DSR in the next two sections .They also gives a very good idea on the advantages of using an On-demand routing protocol. This is followed by the explanation of working of the routing protocol with the help of simulations on how efficiently the protocol works with changes in the density of the node and also when mobility is involved in the network.

The paper "Ad-hoc On-Demand Distance Vector Routing" by C.Perkins and E.M.Royer gives a very good understanding of the AODV routing protocol for Ad-hoc networks. The authors start with the explanation of the importance of route discovery on an on-demand basis in ad-hoc networks since ad-hoc networks are infrastructure less and involve a lot of mobility. The authors also explains the basic operation of the AODV routing protocol which is basically a distance vector routing protocol but route to the destination is discovered only when there is a need for communication between the source and the destination. This is followed by the simulation which gives a very good idea on how the throughput and efficiency is affected under different scenarios like involving mobility of nodes and changing the density of the node.

The research paper "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols" by Josh Broch, David A. Maltz [7] is a very elegant paper that compares the major routing protocols in Adhoc networks on various aspects like mobility, density of the number of nodes and physical and MAC layer behavior of the different Adhoc Routing Protocol. The author gives a very good explanation that each and every protocol designed has a major functionality that's being concentrated like simplicity in the routing, mobility or the density of the node. So routing protocol that's being used in a particular domain basically depends on the major functionality of that domain. So interoperability plays a very important role when there needs to be a communication between different routing domains or across a different routing domain to reach a domain using the same routing protocol.

The paper "An energy based power-aware routing protocol in ad hoc networks"by C.Perkins and S.R.Das [8] gives a very good comparative study of the wireless ad-hoc networks. The author takes two on-demand routing protocols namely Dynamic Source Routing and Ad-hoc On-Demand Routing protocol for study. The gist of the paper is that even though both the routing protocols are based on the same routing principle the performance of the network varies with the load, mobility and the density of the network. It concludes that each of the protocol has its own pros and cons depending on the network design .Thus we can conclude that it would be a hard decision to go for a particular routing protocol leaving the other since each has its own advantage and disadvantage. The better option would be to use the routing protocol that better suits for the network design and implement interoperability between the routing protocols.

The paper "Degree distribution and hop count in wireless ad-hoc networks" by Hekmat, R.Van Mieghem [9] discusses the importance of building an efficient energy aware routing protocol in adhoc networks. The importance of energy aware routing comes into picture since all the nodes in mobile adhoc networks are distributed and the availability of power to the nodes is very limited. Thus it's very much necessary to have an efficient routing algorithm designed so that the energy consumed is being intelligently managed. The author proposes a protocol which uses a "local multicast" mechanism for the discovery of route. This paper also mathematically proves that the energy consumed for the route discovery is considerably reduced by the use of the proposed protocol.

The research paper "Introduction on Ad-hoc Networks" presentation by WAN Chunfeng [10] gives a very good understanding of the calculation of degree of a node and mathematical analysis of calculating the hop count in a mobile ad-hoc network. Since most of the wireless adhoc networks are multi hop and also mobile in nature it is very necessary to know the degree of the node. The Degree of a node is nothing but the number of possible directly connected nodes for any given node. This paper mathematically analyses the calculation of the degree of the node and also the hop count determination in a ad-hoc network

.2.5 Summary

The previous section discussed the basic routing principles of Adhoc Networks followed by the gist of working mechanisms of the three major routing protocols in Adhoc Networks. The above section also explains the pros and cons of each of the routing protocol discussed. From the analysis, it can concluded that each and every protocol has its own pros and cons and the efficiency of choosing the routing protocol for an Ad-Hoc network depends on the type of network being used. The various factors which come into the picture when deciding the protocol include mobility, convergence time for the protocol, throughput, time for Route Discovery and data exchange. There is not a single protocol which satisfies all these requirement of a network. Thus, interoperability between the different routing protocols can be a solution to address this problem. The conclusion and future work are presented in chapter 6.

Chapter 3

Interoperability of Adhoc Routing Protocols

This chapter gives a detailed discussion of the proposed protocol for establishing the interoperability of Adhoc Routing Protocols. This Chapter is organized as follows. Section 3.1 explains the importance of having interoperability with a real time scenario where interoperability between the routing protocols plays a major role. Section 3.2 defines the goals the protocol needs to address, to bring in the interoperability of the routing protocols. Section 3.3 explains the basic algorithm of the protocol and the design principles used in the protocol. Section 3.4 takes in a network scenario to explain the flow of the protocol and how each node reacts to the protocol design.

3.1 Importance of Interoperability of Adhoc Routing Protocol

Interoperability plays a very important role when a routing domain is being separated by another routing domain and there is a need to have a full connectivity in the routing domain even if they are being separated. The best scenario where interoperability of ad-hoc Routing can be applied is disaster recovery. In a disaster recovery scenario there will be different organizations like Military, Medical and Non Government organizations involved. Considering different organizations as different routing domains it is necessary to have communication between them and also between the same organizations which may be separated by a different domain. The interoperability between the routing protocols plays a major role in any other scenario where there are different organizations involved in the ad-hoc Networks. The next section explains the working of the proposed method for interoperability of Adhoc Routing Protocol.

3.2 Goals for the proposed protocol

The first problem to be addressed when designing a protocol to have interoperability between the routing domains in Adhoc Network is to define a universal packet format .This universal packet format is designed to have communication between routing domains to exchange basic information to establish the interoperability between the routing domains. The next goal would be to identify the egress nodes for a particular routing domain .The purpose of identifying the egress node will be to basically decide the node which will be responsible for routing the path outside the routing domain.

Since it is not an efficient design to maintain the routes for the nodes outside the routing domain, this protocol should discover the route to the destination only when the source wants to send information to the destination. This is basically based on the principles of On-Demand Routing in Adhoc Networks. The other factors to be considered when designing the protocol include the following considerations in the design of the protocol The path discovery process should be very efficient, looping of the path should be avoided and the design of the protocol should be simple and efficient so that it does not create much overhead to the functionality of the routing protocol operating within the domain. Considering the above goals that are set for the protocol, the subsequent section explains the universal packet format and the design of the protocol.

3.3 Methodology for interoperability of Adhoc Routing Protocol

3.3.1 Defining the universal packet format

For any routing protocols to communicate with each other there must be a universal packet format which all the routing protocols can understand. This protocol has defined a universal packet format with all the necessary information to achieve the following goals.

- Should discover the neighboring domain.

- Should have knowledge of the egress node to route the packet outside the domain.
- Egress node should have information on reaching the various domains and the information about the ingress node through which the other domain can be reached.

The universal packets are hello messages that are exchanged at regular interval of time. Each and every routing protocol has a distinct route type to determine the routing protocol the node is running. When a node receives a hello packet from a different domain the route type is identified to be distinct from its own .Thus, the nodes exchanging universal hello message from different domain become the egress router for there respective domain.

The proposed universal header format contains the following fields.

- *Route Type:* Identifies the routing protocol used by a particular node when exchanging the hello packet.
- *Code Bits:* Identify the type of packet. The types of packet are Hello Packet, route discovery packet, route reply packet and payload packet.
- *Egress Node ID:* This field in the hello packet identifies the egress node for the particular routing domain and is being populated within their routing domain.
- *Header Length:* Identify the length of the header which helps to point exactly where the payload starts.
- *Source Address:* This is the address of the source node in a given routing domain which is originating the route request.
- *Destination Address:* This will be the address that the source node would like to communicate with at a given instant of time.
- *Request ID:* The unique identity of the route request that is being generated by the egress node is maintained with the help of Request ID.

- *Time to Live:* The maximum time a packet can live. This is basically designed so that the packet does not travel further than the number of hops specified to reach the destination. The length of the Time to Live field is set to 5 bits and thus the maximum number of hops that Route request can travel without getting a reply would be 32
- *Header Checksum:* A checksum on the header only. Since some header fields change (e.g., time to live), this is recomputed and verified at each point that the universal header is processed.
- *Sequence Number:* The sequence number is used to check the freshness of the information received and also to avoid routing loops when interoperability between the routing domains is in operation.

OpCode Value	**Functionality**
001	Exchange of Universal Hello packet
010	Route Request Universal packet
011	Route Reply
100	Data Exchange Universal Packet
111	Destination Unreachable Route Reply

Figure 3.1 OpCode Values for the Universal Packet Format

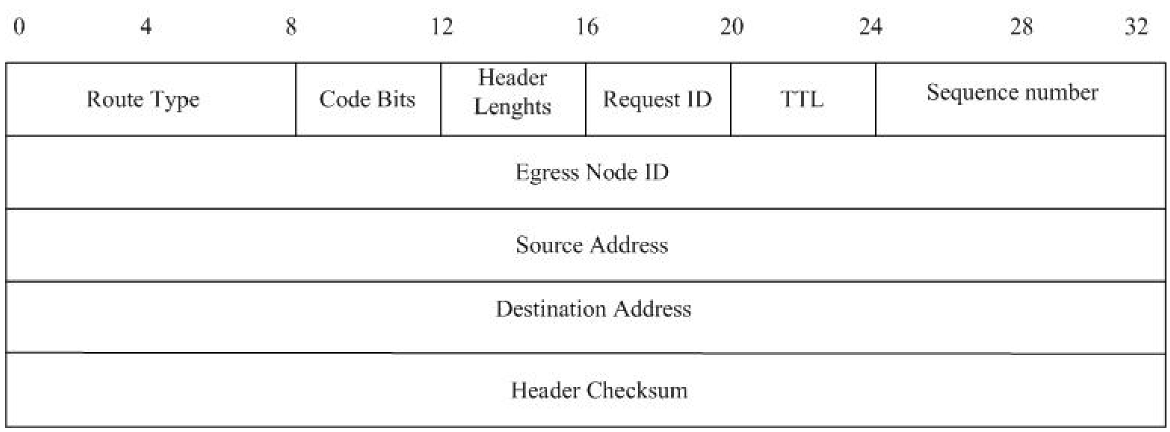

Figure 3.2 Header for Universal Packet Format

From the packet format it can be implied that the overhead involved by the use of the universal packet format is 8 bytes, which is very efficient in bringing interoperability between the Ad-hoc routing protocols.

3.3.2 Egress Table

Along with the routing table each and every node which needs to have interoperability has to maintain an egress table. The egress table basically tells each and every node how to reach the egress point which makes the domain route discovery much faster. The information present in egress table includes:

- *Egress Node*: Information about the node that can reach outside the domain.
- *Domains Connected*: The domains that are connected through the egress node.
- *Dynamic Node determination*: The route to the nodes outside a particular domain that was determined in the previous discovery process. The advantage of having a dynamic node determination is that the route stored in the dynamic node determination column is used for future routing for the particular destination. Aging mechanism can be used for removing the information about any destination which is not being used for a particular length of time.

Having explained the two major requirements for establishing the interoperability of the routing protocols the next section explains the design of the protocol to have interoperability.

3.4 Protocol Design

The protocol design involves three main steps to establish connectivity and have the communication between nodes across the domain.

- *Bootstrap Phase*: This phase is when the node comes up and the egress table is being built.

- *Route Discovery Phase:* In the route discovery phase, the route to the destination outside the domain is determined based on the egress table information and information on how to reach the egress node.
- *Data exchange Phase:* In the data exchange phase, the exchange of data between the source and the destination takes place after the route to the destination is determined. The following subsection will give a detailed overview on the functionality of each and every phase.

3.4.1 Bootstrap Phase

Bootstrapping can be defined as the process of activating simple system level programs for a more complex application which depends on these simple programs to start functioning. In this protocol, for having the more complex route discovery to function, each node should have information about the egress node to be reached so that the route discovery outside the domain can be accomplished. So the goal of the bootstrap phase in this protocol is that each node should have the reachability information of the entire egress node within the domain. To achieve this goal a methodology for identifying the egress node is very important. A node is said to be an egress node only if it can reach a node in another routing domain. The universal packet format discussed in the previous section plays a role in identifying the egress node.

Initially, when the node powers on the universal hello messages are exchanged at regular intervals of time. The opcode value is set to 001 to identify that the functionality of the Universal Packet is to exchange hello messages between the nodes. The *Route Type* field in the universal packet is used to identify to the node if it is exchanging the hello message within the routing domain or with a node in another routing domain. When a node exchanges the universal hello message with a node in some other routing domain, it becomes the egress node for its respective routing domain. Once the egress node is identified, the egress node information is populated to all the other nodes in the domain using the hello packet with its node id in the Egress Node ID field. Each node builds the egress table with the egress node and the information on the domain the egress node can reach. The Bootstrap phase is said to be complete once all the egress nodes for a particular domain are identified. After the synchronization and the completion of the bootstrap process, the Egress Tables of all the nodes within the Routing Domain are supposed to be identical. When the egress table built are identical, the Bootstrap process is said to be complete. The Flowchart in figure 3.2 gives a very good understanding of the concept explained in this section

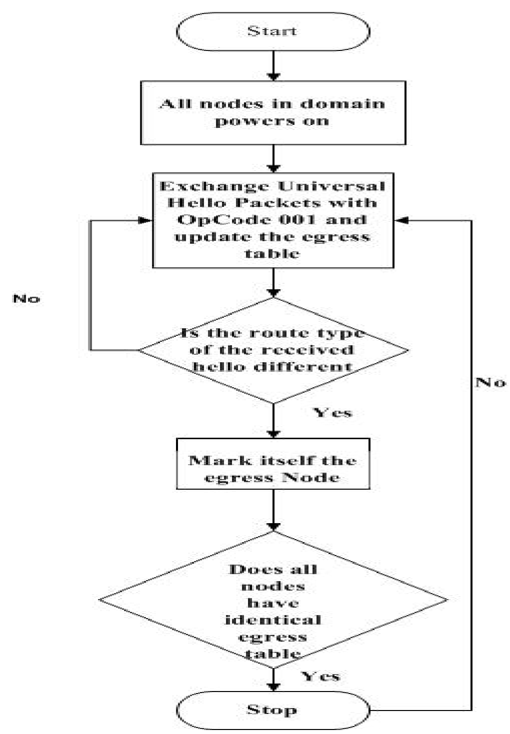

Figure 3.3 Flowchart representation of Bootstrap Phase

3.4.2 Route Discovery Phase

Route Discovery Phase is the second step in this protocol design. This phase is initiated by a node, only if a source wants to start communication with a node which is outside its routing domain. The first step that the node does when it wants to send a packet to its destination is to use its local route discovery process within the routing domain as defined by the routing protocol in use. If it has an entry for the destination, then packet forwarding takes place as defined by the routing protocol used within the domain. If the destination is not within the routing domain, then the route request is sent to all the egress nodes in the domain. The egress nodes in a given domain are determined from the egress table built in the Bootstrap phase. The path determination to reach the egress node is determined by the routing protocol used in the domain. This is done to reduce the complexity of the path determination process. This process is done to all egress nodes within the routing domain. The universal packet being encapsulated has data by the routing protocol and sent to the egress nodes. The source node, after sending the route request to the egress node goes to the active state and waits for a reply from all the egress nodes to which the route request was sent. The egress node on receiving the packet decapsulates it and checks for the destination address. If it finds the route within its routing domain it generates the Route Reply packet and sends to the egress node, initiating the request, else it generates a route discovery packet to be sent to the neighboring domain.

The egress node builds the universal packet format with *opcode* field 010 to identify the packet to be a route request packet. It sends the route request to the node in the neighboring domain with a time to live value set to 63 and the source address as the egress node address originating the request. After sending the route request the node goes into the active state and returns back to the passive state once it gets a reply. The identity of the request sent is maintained using the *Request ID* field in the universal packet format. Once the egress node from the neighboring domain receives the route request it checks if it has entry for the reachability of the destination address. If it has entry for the destination address, then it sends the route reply for the egress node originating the request. If it does not have an entry for the destination node then it identifies all the egress nodes within its routing domain from the egress table and sends the request message to all the egress nodes within the domain. Now the source address in the address field is set to the node id of the local egress node which is forwarding the egress node information to all the nodes in the domain. Also the time to live value is reduced by one and the time to values is reduced has the hop to reach the destination increases. Thus the maximum hop for a reachable destination would be 63.This process of forwarding the route request continues until the destination is reached or the route to the destination is determined.

Once the route to a destination is determined by the intermediate node or the destination node itself, the route reply packet is sent back to the node originating the request. The Route Reply packets are identified with the Opcode value 011 .When the egress node receives the route reply for the destination address, it goes back to the passive state. The number of hops to receive the route reply becomes the metric for the egress node. The number of hops is calculated from the time to live field in the route reply. To make the protocol simple and abstract, the egress node sends a route reply to the source node originating the request with only the information of the reachability of the destination and the metric with which it can reach the destination.

If the node is unreachable by the egress node, i.e. if it takes more than 63 hops to reach the destination then the egress node sends a route reply with a flag saying that the destination is unreachable. Once the source node receives replies from all the egress nodes within the routing domain it goes back to the passive state. When the source node originating the request goes back to the passive state the route discovery process is said to be complete. The Flowchart in figure 3.3 gives a diagrammatic explanation of the Route Discovery Phase

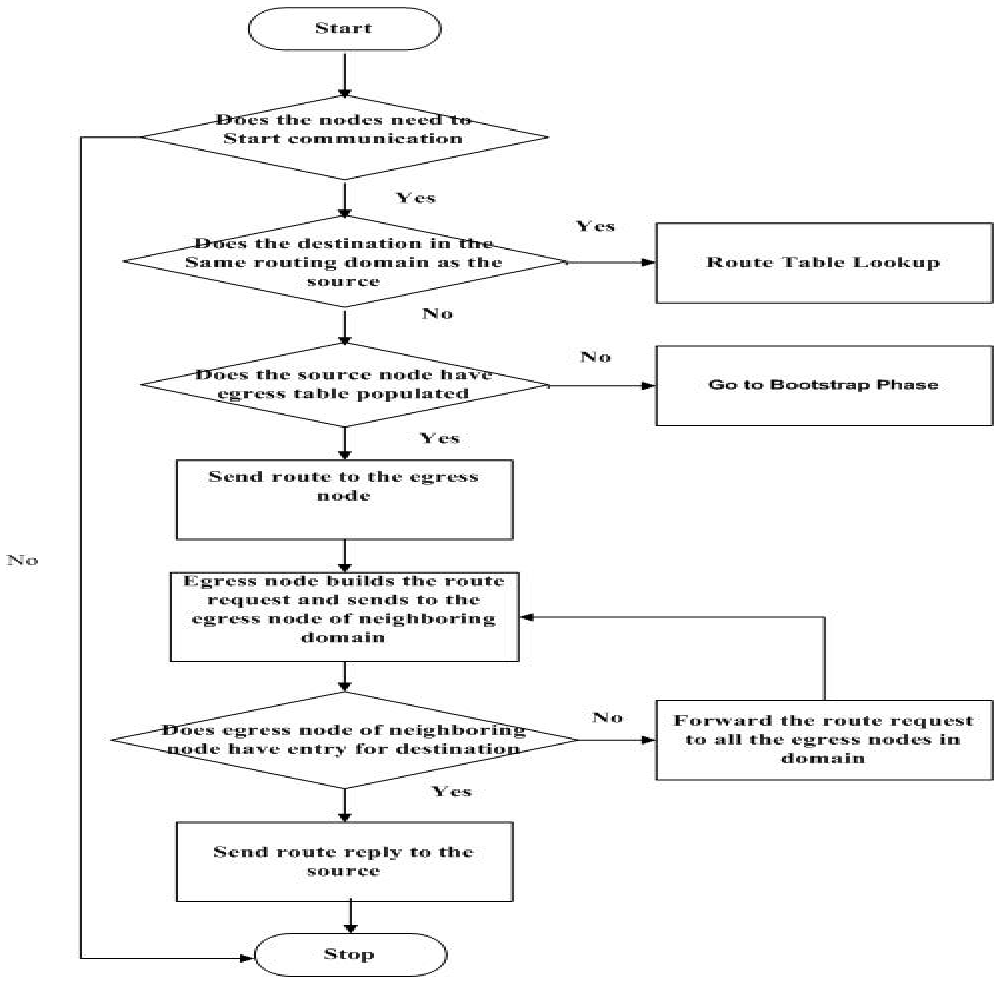

Figure 3.4 Flowchart representation of Route Discovery Phase

3.4.3 Data Exchange phase

After the route to a destination is determined, the next step is the actual exchange of data between the source and the destination. In the third phase, the data exchange begins by comparing the metric that is advertised by the egress node. Then, the source decides the most efficient path from the advertised metrics and forwards the packet to the egress node. The source has no information on the path taken by the data to the destination. Every node forwarding the packet only knows the next node that will receive the packet. The path information is kept abstract to reduce the complexity of the algorithm.

When the egress node receives the data with the destination address outside the domain it matches the request id to identify the node to which it needs to forward the packet. The Request ID in the universal packet format is the unique value for each and every communication to identify the next hop node that will receive the packet. The egress node encapsulates the data in the universal packet and sends it to the egress node of the neighboring domain. The payload packet is identified in the universal packet by the *Opcode* Value 100.Once the egress node receives the packet, it identifies the egress node within its domain to which the packet needs to be forwarded and forwards the packet to that egress node. The path determination to reach the egress node within the domain is based on the routing protocol used within the domain. This is again done to make the approach simpler. As the complexity of the algorithm increases, it results in slower convergence and slower rate of data transfer. Once the egress node receives the data it again encapsulates the data in universal packet format and forwards to the next hop node determined from the unique request id for the destination. This process continues until the destination is reached. Thus, by this protocol, there can be communication established between any two routing domains and interoperability can be achieved. The figure 3.5 gives a very good picture of the Data Exchange Phase.

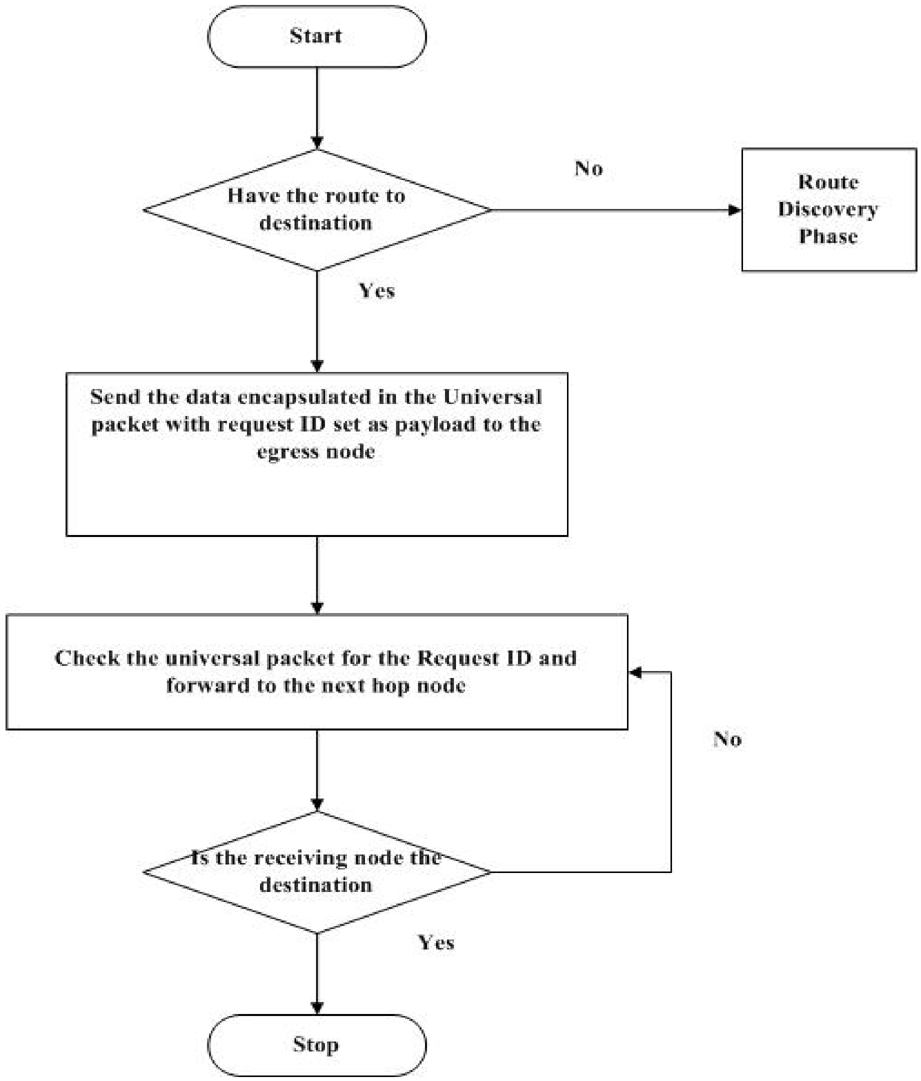

Figure 3.5 Flowchart representation of Data Exchange Phase

Scenario Example

In this section based on the protocol explained in the previous section, an example scenario is being used to explain the working of the protocol to have interoperability between two routing domains.

Assumptions in the Scenario

- All the nodes are static and there is no mobility involved in the scenario.

- Each and every node understands the universal packet format that was explained in the previous section.
- All the nodes in a given Routing Domain run the same routing protocol which is different from the routing protocol being run in the neighboring domain. Each routing domain works on a standardized routing protocol which has a route type that is defined in the Universal Packet Type.

In the scenario given below the goal is to have communication between A1 and A5 which are in the same routing domain but are separated by another routing domain B as shown in the figure 3.6

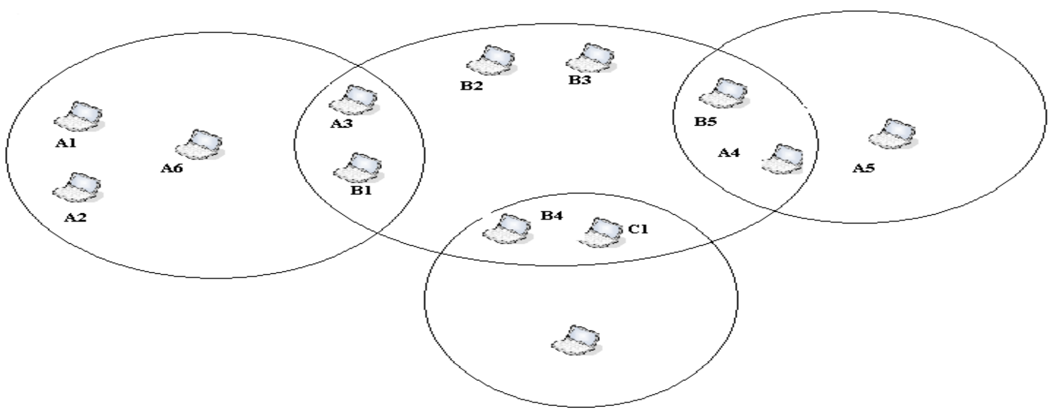

Figure 3.6 Scenario for the working of Interoperability of Ad-hoc routing protocols

The explanation of the scenario is divided into three phases as explained in the previous section.

Boot Strap Phase

After coming up, all the nodes in domain A, B and C start exchanging the universal hello packet of *opcode* type 001.When A3 broadcast its hello packet, B1 receives it with a different route type and thus identifies itself as an egress node. In the same way when A3 receives the hello message from a node in domain B it identifies itself as egress node for domain A.

The node A3 then populates all the nodes in its domain by advertising itself to be the egress node. Similar operation happens in all the domains and identification of the egress node and building of the egress table happens in the first phase. The snapshot of the egress table of Domain A and B are given in the table below.

Egress Node ID	**Domain Connected**	**Dynamic Node Determination**
A3	B	

Egress Node ID	**Domain Connected**	**Dynamic Node Determination**
B1	A	
B6	C	
B4	A	A5

Figure 3.7: Egress Table of A before route discovery

Route Discovery Phase

This section explains how the route discovery process exactly takes place when A1 wants to communicate with A5.

First when the source (A1) wants to communicate with the destination (A5) it checks its routing table to find if it has entry for the destination or initiates the route discovery process within the domain to find if A5 is within the reachability of A1.In the given scenario since A5 is not within the domain, the route discovery function within the domain fails and the source goes onto the next step.

Node A1 checks its egress table to identify the egress nodes within its domain and forwards the route request message to A3 and goes into the active state. When A3 receives the route request it builds the Route Request Universal Packet with opcode type 010 and sends to B1.In the route request packet it generates a unique request id for this particular route request and sends it to B1 as shown in the figure.

B1, on receiving the route request, checks its routing table to determine whether it has an entry for A5.Since it does not have entry for A5, it encapsulates the route request as a data into its routing protocol header and sends it to the entire entire egress node within the domain. On receiving the route request, all the egress nodes continue to forward the route request to the egress node in the adjacent domain. This process continues until the destination is reached or the route to the destination is found in any of the intermediate nodes. Each node goes into the active state after sending the route request to the next node.

When the route request reaches the destination node it builds the route reply packet and sends the route reply packet to the egress node within its domain which made the route request. The node which made the route request is determined in the return path using the unique request id in the route request packet. Each node on receiving the route reply keeps following the path back to reach the destination. Once the route reply is received the node goes back to the passive state. When the route reply reaches the final destination A1, it makes the entry in its egress table in dynamic node determination field and goes back to the passive state as shown in the figure 3.Thus A1 knows A5 can be reached by forwarding the packet to A3.This abstraction in path information is made to reduce the complexity of the algorithm.

Egress node	Domain connected	Dynamic Node Determination
A3	B	A5

Figure 3.9: Egress Table of A after route discovery

Data Exchange Phase

After the first two phases the data exchange phase is fairly simple. A1 sends the data with the universal header, which becomes the payload for the routing protocol within the domain to the egress node. The egress node matches the destination address and the Request ID and finds that the next hop node is B1 and forwards the packet to B1. B1, on receiving the packet with the unique request id, identifies the packet needs to be forwarded to B4 and sends the packet to B4.This process is continued until the destination is reached. Once A5 gets the packet, it strips off the packet header and the universal header and receives the data from A1.Thus data exchange is achieved between the source A1 and destination A5.For any further communication to A5 from any of the nodes in the domain .A egress node A3 matches the destination address and the request id and sends the Route Reply back to the requesting node. This process reduces the route discovery time for A5 from any of the nodes in domain A.

Step 1:A1 checks its routing table build by the routing protocol used by A and finds that there is no entry for A5.

Step2: A1 checks its egress table and sends a route request to the egress node A3 and will be on hold for a particular hold down time to get a response.

Step 3: A3 encapsulates the route request in the universal packet and sends to B1.

Step 4:B1 from its egress table knows that the probability of reaching the domain A on the other side is more through the egress node and sends the route request to B4. (This request is sent in the format that the routing protocol understands. This is not done using the universal packet format)

Step5: B4 encapsulates the destination address in universal packet and sends it A4.

Step6: A4 knows the reachability of A5 and sends a reply back to B4.This process continues until A1 and now A1 will have the reachability for A5.

This information is being recorded in the respective egress table for future communication.

Chapter 4

Mathematical Model and Simulations

4.1 Introduction

In this chapter a mathematical model is designed to find the time taken in Bootstrap Phase and the Route Discovery Phase. This will be followed by the discussion on the simulation environment and the results. The conclusions of the results are discussed in the last section where the analyses of the results obtained under different scenarios are being made.

In the first section of this chapter a mathematical model is proposed to determine the time taken for the Bootstrap phase. Bootstrap Phase is when the all the node boots up and starts the exchange of hello messages .Through the exchange of hello messages, the egress nodes in the domain are being determined as discussed in the previous chapter. This section analyzes the time involved for the egress node information to be populated to all the nodes in the routing domain.

C++ is used to program the designed algorithm to determine the time involved for the egress node information to be populated to all the nodes in the routing domain. Five different scenarios and analyze how the time involved in the egress node information to be populated varies as the depth of the node increase from the egress node. The output from program is used in MATLAB to graphically show the results. This chapter is organized in the following manner. Section 4.2 explains the Mathematical Model for the Bootstrap Phase. This section starts with the calculation of the degree of the node followed by the algorithmic explanation of the time calculation in the Bootstrap phase. This is followed by considering an example of a scenario in explaining the algorithm. In section 4.3, we give a detailed explanation for the time calculation involved in the Route Discovery Phase. This section starts with the assumptions that we make for the time calculation involved in the Route Discovery Phase followed by the mathematical model involved in the time calculation of Route Discovery Phase. Section 4.4 discusses the simulation for the time calculation in the Bootstrap Phase and the Route discovery phase. In section 4.5 the author analyses the results obtained from the simulation and explains the inferences from the simulation. Section 4.6 summarizes the chapter in which the author gives a brief explanation on the mathematical model and simulation discussed in this chapter.

4.2 Mathematical analysis of Bootstrap Phase Time

As discussed in the previous chapter the main goal of the protocol in the Bootstrap Phase is that all the nodes in a given routing domain should know how to reach the egress nodes so that they can reach the outside world.

Egress Node Information Time can be defined as the time taken for all the nodes in a given domain to learn about the Egress nodes. The modeling of wireless ad-hoc network can be represented by a geometric random graph denoted as $G_{p(dab)}(n)$ where P(dab) is the probability of having a link between a and b separated by a distance of dab from each other.

4.2.1 Determination of the degree of a node

In the following subsection of the thesis explains the requirement for the determination of the degree of a node for the maximum time calculation in each depth .The degree of the node is defined as the number of links [10][14] in the transmission range of the given node in a given area. If n nodes are uniformly distributed in a given routing domain then the degree of a node can be defined by equation (1)

$$\mathrm{E}\left[d\right] = (n-1)\eta \qquad (1)$$

where η the link density is defined as

$$\eta = \frac{2}{n(n-1)}\sum_{i=1}^{m}\sum_{j=i+1}^{m} p(r_{ij}) \qquad (2)$$

Where m is the square units covering the area of the routing domain in such a way that each m square can utmost hold only one node [14].

Having known the degree of a particular node this mathematical model now formulates the time for the Bootstrap Phase .This mathematical model is based on a pessimistic approach where we find the maximum time that would be taken for a Bootstrap Phase to be completed in a particular domain. An example of a scenario would help us to go with the Mathematical Formulation.

4.2.2 Algorithm for the Time Calculation in Bootstrap Phase

The Boot Strap phase basically consists of populating the egress node information to all the nodes in a given routing domain. This formulation of the time calculation determines the maximum time taken for the Bootstrap Phase to be completed. The depth of each node is a tree representation of the number of hops from the egress node where the root node of the tree always represents the egress node.

The algorithm for the mathematical model works in the following method. It start with the time calculation from the highest depth and continue until we reach the root node.

Step 1: Each and every node in a given depth stores the time T_i where i represents the depth from the root node. The variable T_i is the time taken by a particular node to reach the node in the last depth.

Step 2: When 2 nodes are in the same depth and are connected to the common parent the node with the maximum value for T_i sends the time value to the parent node. This is basically done to have the maximum time involved in the Bootstrap Phase. Thus the time involved in the Bootstrap phase can be lesser than the calculated time but not more than the time calculated.

Step 3: As the value of i increases the time value of the node with maximum value of T_i is being sent to the upper parent node. This iteration continues till the value of i becomes 1.

Step 4: Thus if T_j is the time taken for the egress node information to be populated to all the nodes in a given domain then T_j can be represented has follows.

$$T_j = \max_{j=1}^{\max(d)} T_i + C_t \qquad (4)$$

Where C_t represents the constant time that's involved when all the nodes come up and basic Bootstrap process of all the nodes take place.

4.2.3Example Scenario for the Mathematical Model

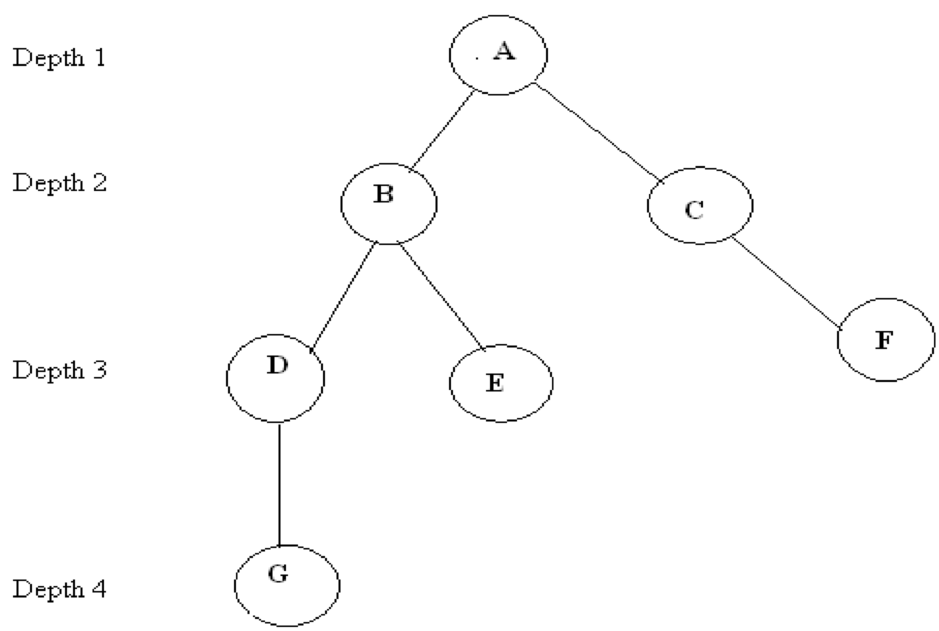

Figure 4.1 Scenario for the Mathematical model in Bootstrap Phase

Figure 4.1 represents a tree structure of a wireless ad hoc network. The tree structure represents an adhoc network in which A will be the egress node and the directly connected children of any node represent the nodes within the broadcast domain of the node. For example B is a node which are in the broadcast range of A ,D and E.C is in the Broadcast range of A and

F. This degree of a node can be determined by the formulation discussed above in equation (4).Having determined the degree of a node the next step will be to calculate the time taken for the Bootstrap Phase.

As per the algorithm explained in the previous section the depth of the domain will be 4.Starting from the depth 4 we check for the maximum time as we go up the tree towards the root node.

Step 1: The first step would be for the node G and it would check if it has any child. Since it does not have any child it will have the value of T_i set to the time required to reach its parent i.e. node D and send it to node D.

Step 2: Now the value of the time T_i is compared between D and E in the depth 3.This is done because they belong to the same depth from the root node and also have a common parent B.Since the value of T_i in node D is greater that the value of T_i in node E, node D adds the value of the time taken to reach B and sends its parent i.e. node B.This new value for T_i is stored in B.In the same manner F sends a value T_i to its parent C where T_i is the time for transmission between F and C.

Step 3: The same iteration continues for depth 2 where B and C compares the value of T_i since both the nodes belong to the same parent and positioned in the same depth from the root node.

Thus it's being observed from the working of the algorithm, it is basically based on the equation discussed in the previous section. In this scenario the equation would be

$$T_j = \underset{i=1}{\overset{d(\max)=4}{MAX}} T_i + C_t$$

4.3 Time for the Route Discovery Phase

As discussed in the previous chapter route discovery phase is when the source needs to find a route for the destination and start the exchange of data between the source and the destination. The goal of this section is to mathematically determine the time taken for the route discovery process. The methodology for the mathematical calculation of the route is a pessimistic approach in which the maximum time is determined for the route discovery on a given scenario. The route discovery time can be lesser than the value derived from the mathematical model but not greater than the derived results.

4.3.1 Assumptions for the Time calculations in the Route Discovery Phase

- The area and the dimensions of a given domain do not change at any point of time after the Bootstrap Phase.
- The nodes are assumed to be immobile and do not move from one domain to the other when they are in the route discovery phase.
- The time required for the transmission of universal packet over a unit area is assumed to be a constant across the routing domain.

4.3.2 Time calculation for the Route Discovery Phase

To determine the time involved in the route discovery process let us consider the scenario given below.

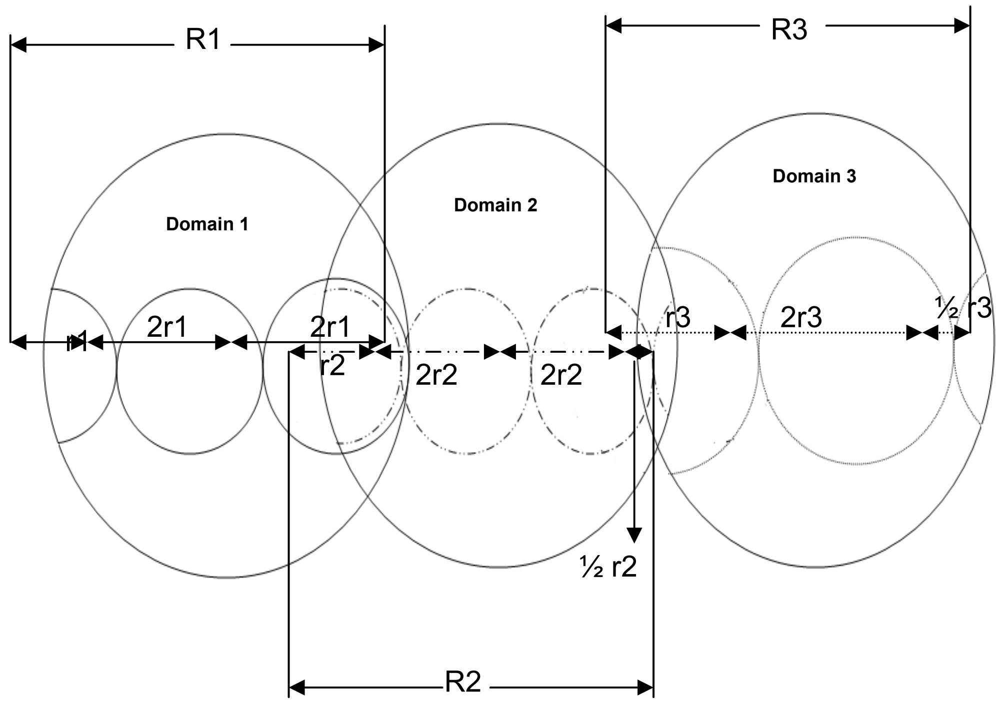

Figure 4.2 Pictorial representation for time calculation in Route Discovery Phase

A scenario of the route discovery involved from a node in Domain 1 to a node in Domain 3 is being considered .The time required for transmission of universal packet across a unit area is t_u. The broadcast radius for a domain n is considered to be r_n and the radius of the domain n is considered to be R_n.

Thus the unit area in the broadcast domain for the domain n can be expressed as follows:

Unit area in the domain n = $\frac{rn}{Rn}$ (5)

As defined earlier the time required for transmission over a unit area is t_u , the time required to transmit a universal packet across one domain can be given as

$T_{RDi} = t_u \bullet \frac{rn}{Rn}$ (6)

This will be the maximum time required for a node to transmit a universal packet across the domain n.

If the route discovery process involves the transmission of the universal packet across a domain by the egress node, then the time required for transmission across the second domain is added to the time required to transmit the universal process across the first domain. In general, if the route discovery involves the transmission of universal packet across n domains to reach the destination, then the time for transmission of the universal packet can be given as

$$T_{RDN} = tu \bullet \left[\frac{r1}{R1} + \frac{r2}{R2} + \frac{r3}{R3} \ldots \frac{rn}{Rn} \right] + Tc \qquad (7)$$

The constant time Tc involved is due to the time required when an egress node needs to exchange the universal packet with the egress node of the neighboring domain. There is a time involved for this because the egress node needs to strip off its header and just send the universal packet to the egress node of the neighboring domain.

The total time required for the source to know the route discovered to the destination will be the round trip time involved for the route discovery which can be expressed as follows.

$$T_{RD\,N} = 2\, tu \bullet \left[\frac{r1}{R1} + \frac{r2}{R2} + \frac{r3}{R3} \ldots \frac{rn}{Rn} \right] + Tc \qquad (8)$$

4.4 Results and Analysis

This section of the thesis discusses the simulation environment and the results. This will be followed by the conclusion of the results where the analyses of the results obtained under different scenarios are being made.

4.4.1 Goal of the Simulation

Having derived the mathematical equation for the Bootstrap Phase and Route Discovery Phase, the goal of this simulation would be to program the equation to get the results under different scenarios. The different scenarios being defined are based on the variation in depth of the domain from the egress node. This would give us a clear understanding of the how the time involved in the Bootstrap Phase would be affected by changing the depth of the node.

C++ programming language is used for the simulation to find the time taken for the Bootstrap phase at different scenarios where the depth from the root node i.e. the egress node changes. The inputs given to the program are the number of nodes, name of each node and the parent of the node. Having given the following inputs it has been programmed to find the time for the Bootstrap phase using the equation discussed in the previous chapter.

4.4.2 Assumptions for the simulation

- The nodes which are being considered for the time determination in the Bootstrap phase are immobile.
- Each node on a given routing domain knows the nodes in its broadcast range by the exchange of the universal hello packet. .
- The value of the constant C_t is assumed to be the same for all the nodes in the routing domain.

4.4.3 Explanation of the simulation program for Bootstrap Phase

The program starts by defining the various variables that are going to be used in the simulation. Then a structure "Nodes" is being defined which is used to define the attributes of each and every node in a given scenario environment. The program takes in three inputs from the user namely Node Name, Parent node and Distance to the parent node.

Node Name: The variable node name is used to identify the node for which the attributes are being defined.

Parent Node: The parent node is defined in a structure to define the name of parent node for a given node. For an egress node which is assumed to be the root node the parent node is declared to be null.

Distance to the Parent: The distance of a node from the parent is being defined in this variable. With the distance and the parent node as the input it is been programmed to automatically design the tree structure to find the distance to the root node.

Given the inputs it is programmed to determine the distance from the root node to each and every node. Having computed the distance to all the nodes the next step would be to calculate the time taken to reach a node from a particular depth. Since the time for communicating with the next hop router depends on factors like the size of the network environment, physical layer and MAC layer properties of the network, its assumed the standard broadcast time for one packet to be transmitted to the next hope node to be 1 second. The next step in the program is to use the mathematical equation discussed in the previous chapter to determine the time taken to reach the root node by all the nodes which is nothing but the time taken for populating the egress node information to all the nodes in the routing domain. We start from the highest depth and go in ascending order and find the maximum time in each depth which is being sent to the parent node. This process continues until the root node is reached. The time value in the root node gives the maximum time required for populating the egress node information in the given domain.

4.4.4 Scenarios of Simulation

In this section, four scenarios are taken into consideration and analyze the maximum time in the Bootstrap phase in each scenario by changing the depth of the tree structure as shown in the figure

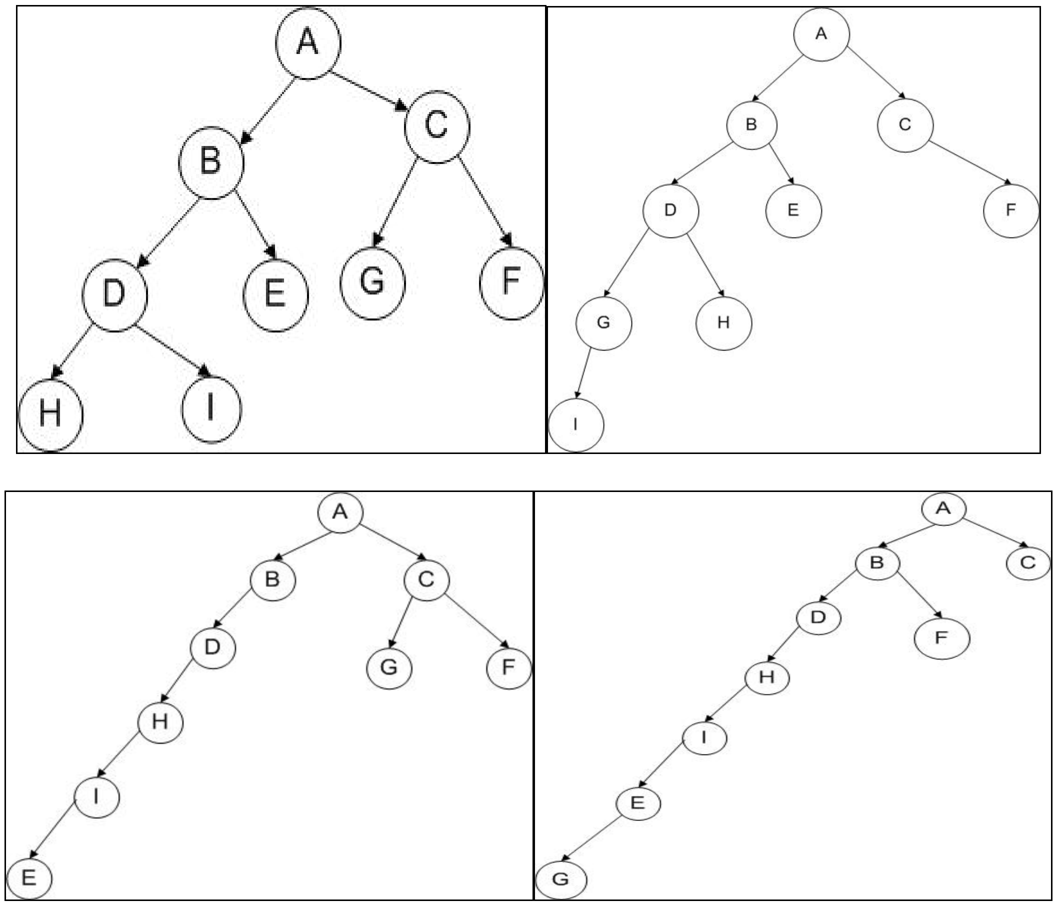

Figure *4.3*: Tree structure of Scenario 1,2 3 and 4 in clockwise direction

Graphically representations of the result with the help of MATLAB are being used. In all the scenarios, the numbers of nodes are assumed to be 9.The simulation is started by having the depth set to3 and then increase the depth in each scenario and tabulate the outputs that we get.

The inputs are given to the C++ program based on figure 4.3 .The broadcast distance is defined in the program and the tree structure is calculated based on the defined broadcast distance. The value of T_j will be the maximum value when compared between the product of the distance advertised by its children and the time to reach the root node.

This iteration continues down the depth by increasing the depth value and comparing the time at that particular depth. In the scenarios given in figure 4.3, the network is being designed for analysis in such a way that the tree structure in scenario 1, 2, 3 and 4 has a depth of 3, 4, 5 and 6 respectively.

4.4.5 Explanation of the simulation program for Route Discovery Phase

In the next simulation program in C++ we analyze the different scenarios that will affect the route discovery time in route discovery phase. The two main parameters involved in route discovery phase are the domain size and the broadcast range. This thesis takes the different scenarios by changing the domain size and the broadcast range and analysis how it affects the route discovery time. The simulation consist of a C++ program which takes the number of domains involved in the network as input and the broadcast range in each domain. In the simulation the C++ program creates an array depending on the number of domains in the network. The program gets user input on the values for the domain size and the transmission range. The simulation then uses the mathematical model being designed for the route discovery phase to find the time involved in the route discovery phase. In the first set of the simulation involves varying of the size of the domain and analyze how the route discovery time is affected. In the second part of this simulation varies the transmission range in each domain and examine the results based on the result. The value of the transmission range is varied from 100 meters to 200 meters. This value is being used because this is the most common transmission range for outdoor low-gain antennas [15] which are placed 1.5m above the ground surface and operating in the 1–2GHz band varies from 100meteres to 200 meters.

4.5 Results

4.5.1 Inference of results from Bootstrap Phase

The results are obtained by plotting the graph in MATLAB with the output obtained from the C++ program. From the results, it can be inferred that as the depth increases the value of T_j

also increases. This can be inferred from the graph shown in figure 4.4 in which the x-axis represents the value of depth that was taken for analysis, and the y-axis is plotted for the time from each scenario.

Also it can be inferred from the graph that the time value does not increase proportionately with the increase in depth .The amount of time increase at each depth depends on the distance between the parent and the child at that level of depth. The distance between the parent and child is nothing but the distance between any two nodes in the direction towards the egress node. Thus, from the results, it can be inferred that the time in the Bootstrap Phase mainly depends on the two factors namely the depth which is nothing but the number of hops the source is away from the egress node and the second factor being distance between the parent and child which can be defined as the distance between two nodes where the node nearer to the egress node becomes the parent and the node which is in its broadcast domain of the parent becomes the child.

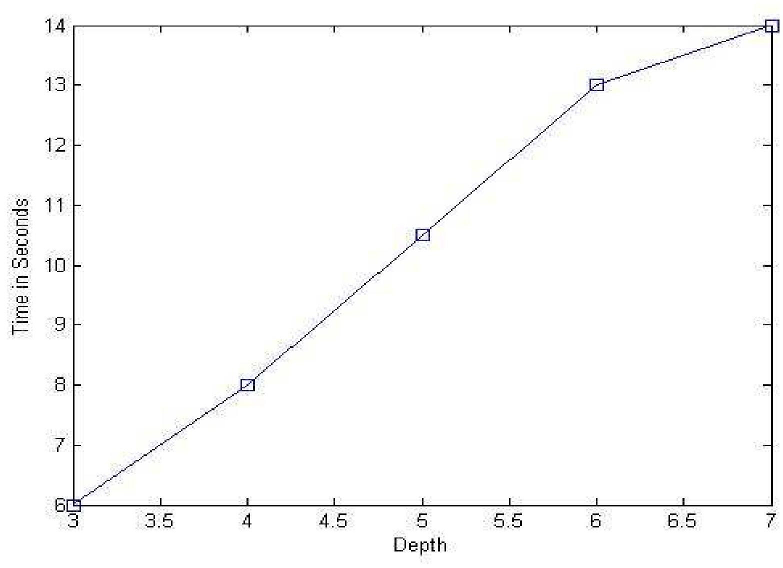

Figure 4.4 Simulation results graph for Bootstrap Phase

4.5.2 Inference of results from Route Discovery Phase

MATLAB is used to plot the graph from the results obtained from the C++ program. It can be inferred from the graph in figure 4.5 that as the size of the domain increases the time required for the route discovery also increases .In the first scenario of the simulation the transmission range is kept constant to 120 meters and the analysis is done by varying the size of the domain. Since the transmission range is constant across the domain and as the domain size increases the number of hops required to reach the destination from the source increases resulting in the increase in the time required for the route discovery phase.

In figure 4.6 the author analyses the variation of the route discovery time based on the changes in the value of the transmission time having the size of the domain constant.

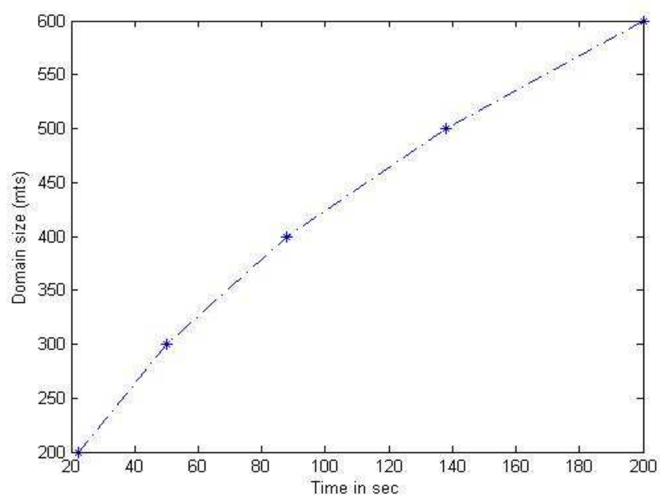

Figure 4.5 Size of Domain vs Route discovery time

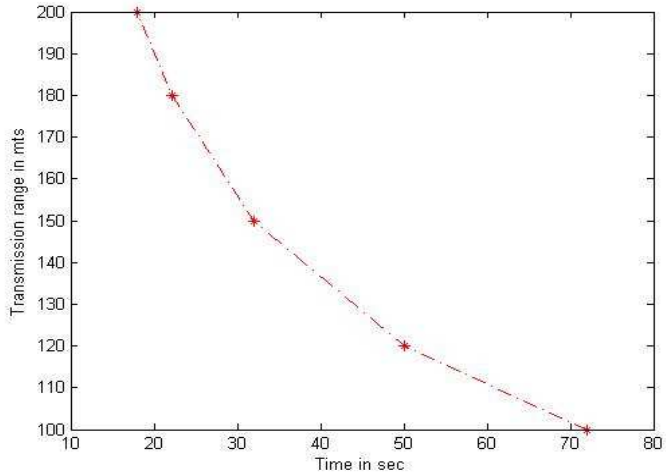

Figure 4.6 Transmission Range vs Route discovery time

It can be inferred from the analyses by changing the transmission range keeping the domain size constant that as the transmission range increases the time involved in the route discovery process decreases. This is because as the time transmission range increases the number of hops to reach the destination will decrease thus the time required for the route discovery phase decreases as the transmission range increases.

4.6 Summary

This chapter of the thesis explains the mathematical model for time calculation in two phases namely the Bootstrap Phase and the Route Discovery Phase .In the first part of the explanation about the time calculation for the Bootstrap phase starts with the mathematical expression for determining the degree of the node from which the maximum time involved in Bootstrap Phase is determined. In the next section of the explanation discusses the maximum time involved in the Route Discovery phase. The mathematical equations are a pessimistic approach which gives the maximum time that would be involved in each phase. But the time involved could be less than the value from the mathematical expressions but not more than the value from the derived expressions.

This is followed by the discussion on the goals of our simulation .It also discussed about the various scenarios taken into consideration and the last section gave the analysis of the results obtained from the simulation.

Chapter 5

Conclusion and Future Work

6.1 Conclusion

This thesis basically starts with an introduction to wireless ad-hoc networks and a literature survey which analyses the work done in the field and the current challenges that of ad-hoc networks faces. After analyzing the research work done in the field of ad-hoc networks, the thesis discusses the importance of having interoperability of ad-hoc routing protocols and a major research areas that needs to be concentrated. This thesis report also gives a real time example of the best place where the importance of interoperability of ad-hoc routing protocols can be felt.

This thesis gives a novel approach by defining a universal packet format which would be understood by all the nodes running all routing protocols. It also gives a very good explanation by dividing the protocol working for having interoperability to three major phases namely Bootstrap phase, Route discovery Phase and Data exchange phase.

Thus, dividing the design of interoperability in adhoc routing protocols into three major phases, this thesis goes on by taking an example scenario on how the interoperability will work explaining each phase related to the example discussed.

In the next chapter a mathematical analysis of the maximum time that would take in the Bootstrap Phase and the Route Discovery phase. C++ programming language is used to program the time calculation for the Bootstrap phase from the mathematical equation being defined for the Bootstrap phase.

Thus, this thesis presents a very important research area in the field of ad-hoc networks which needs to be concentrated on having interoperability between the ad-hoc routing protocols.

Having known the properties of ad-hoc networks such as mobility, infrastructure less networks and self forming networks, it would be a major breakthrough in field of wireless network if there is interoperability between the ad-hoc networks running on different routing domains.

6.2 Future Work

Having proposed the protocol and mathematically modeled the time involved in the Bootstrap Phase and the Route Discovery Phase the next step for future work would be to implement the protocol in a real time environment which consist of nodes running different routing protocols and check the performance of the protocol under different scenarios.

The other future work for this thesis would be to calculate the exact time involved for the constant Ct in equation 4 and 8 defined for the Bootstrap phase and the Route Discovery phase.

REFERENCES

REFERENCES

[1] "Mobile Ad-Hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations". S. Corson and J. Macker, IETF RFC 2501, January 1999.

[2] **"**A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks"Elizabeth M. Royer, Chai-Keong Toh, June 1999

[3] "Routing in ad hoc networks of mobile hosts" Johnson, D.B. Mobile Computing Systems and Applications, 1994. Proceedings, Workshop on Volume, Issue, 8-9 Dec 1994

[4] "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers" by C.Perkins and P.Bhagawat, June 1994

[5] "Dynamic Source Routing in Ad Hoc Wireless Networks" by David B. Johnson David A. Maltz, June 1996

[6] "Ad-hoc On-Demand Distance Vector Routing "by C.Perkins and E.M.Royer, June 1999

[7] "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols" by Josh Broch, David A. Maltz, David B. Johnson Mobile Computing and Networking, June 2000

[8] "An energy based power-aware routing protocol in ad hoc networks" by Jing Nie; Zheng Zhou Communications and Information Technology, 2004. ISCIT 2004. IEEE International Symposium on Volume 1, Issue, 26-29 54

[9] "Degree distribution and hop count in wireless ad-hoc networks" by Hekmat, R.Van Mieghem Networks, 2003. ICON2003. The 11th IEEE International Conference .

[10] "Introduction on Ad-hoc Networks" presentation by WAN Chunfeng

[11]Andrew S. Tanenbaum, .Computer Networks, Third Edition, Prentice Hall of India, October 2000.

[12] "Scalable Routing Strategies for Ad Hoc Wireless Networks." Atsushi Iwata, Ching-Chuan Chiang and Chen,, IEEE Journal on Selected Areas in Communications, Special Issue on Ad-Hoc Networks

[13]"Ad hoc On-Demand Distance Vector (AODV) Routing" C.Perkins and E. Belding-Royer IETF RFC 3561 July 2003.

[14]"Mathematical Energy Consumption Model for Performance Analysis of On-Demand Ad-Hoc Routing Protocol" a thesis report by Rajesh Kumar Rauniyar Wichita State University, December 2004.

APPENDIX

APPENDIX A

A1. Simulation for Bootstrap Phase Time determination

The following code in C++ is used for the simulation of the time determination in the Bootstrap Phase.

Simulation Program

```
#include "iostream.h"
#include <stdlib.h>
main(int argc, char* argv[])
{
        int findNodeNumber;
        int totalNodes;    // input of the total number of nodes
        char findNode;
        int tempNode;
        float currentLength;
        char choice;
        char prevNode;
        int i, j, k;
        float timePerDepth;
        timePerDepth=2;
        cout<<"Enter the total number of nodes:"; //input from the user on the number of
                                                         nodes
        cin>>totalNodes;
        struct Nodes
        {
                char nodeName;
                int root;
                float len;
        }node[20];
        node[1].nodeName = 'A';
        node[1].root = 'O';
        node[1].len = 0;
        for (i=2; i<=totalNodes; i++)
        {
                cout<<"Enter details for node "<<i<<":\n";
                cout<<"Node Name:";
                cin>>node[i].nodeName;
                cout<<"Root Node:";
                cin>>prevNode;
                k=1;
                while (node[k].nodeName != prevNode)
                {
```

```
            k++;
            }
            node[i].root=k;
            cout<<"Length:";
            cin>>node[i].len;
            cout<<"=========================\n";        }
        nodeSearch:
        currentLength = 0;
        cout<<"Enter the node to find the depth and time:";
        cin>>findNode;
        for (j=1; j<=totalNodes; j++)
        {
            if (node[j].nodeName == findNode)
                findNodeNumber = j;
        }
        tempNode = findNodeNumber;
        while(node[tempNode].root != 'O')
        {
            currentLength = currentLength + node[tempNode].len;
            tempNode = node[tempNode].root; }
        cout<<"Depth of the node "<<findNode<<" is : "<<currentLength<<"\n";
        float totalTime = currentLength*timePerDepth;
        cout<<"Time taken is :"<<totalTime<<"\n";
        cout<<"Do you want to search another(y/n)";
        cin>>choice;
        if (choice == 'y' || choice == 'Y')
        {
            cout<<"=========================\n";
            goto nodeSearch; } }
```

Simulation for Route Discovery Time determination
The following code in C++ is used for the simulation of the time determination in the Route Discovery Phase.
Simulation Program

```
#include "iostream.h"
#include <stdlib.h>
#include <conio.h>
main(int argc, char* argv[])
{
        float n, tu, R[100], r[100], TfR;
        system("cls");
      cout<<"Enter the Number of domains in the Network:";
      cin>>n;
      cout<<"Enter the value for time tu:";
      cin>>tu;
        for (int i = 1; i <= n; i++)
```

```
    {
        cout<<"Enter the Domain Size R for the domain"<<i<<":";
        cin>>R[i];
        cout<<"Enter the Broadcast range for the domain"<<i<<":";
            cin>>r[i];
    }
    TfR = 0;
    for (int j = 1; j <= n; j++)
    {
            TfR  = TfR + ( (R[j]*R[j]) / (r[j]*r[j]) );
    }
    TfR = 2*tu*TfR;
    cout<<"Time for Route Discovery is "<<TfR;
    getch();
}
```