UNLV | University Libraries
University of Nevada, Las Vegas

UNLV Retrospective Theses & Dissertations

1-1-2008

Efficient broadcasting by selective forwarDing

Balaji ashok Sathyanarayanan
University of Nevada, Las Vegas

Follow this and additional works at: https://digitalscholarship.unlv.edu/rtds

Repository Citation

Sathyanarayanan, Balaji ashok, "Efficient broadcasting by selective forwarDing" (2008). *UNLV Retrospective Theses & Dissertations*. 2322.
https://digitalscholarship.unlv.edu/rtds/2322

This Thesis is brought to you for free and open access by Digital Scholarship@UNLV. It has been accepted for inclusion in UNLV Retrospective Theses & Dissertations by an authorized administrator of Digital Scholarship@UNLV. For more information, please contact digitalscholarship@unlv.edu.

EFFICIENT BROADCASTING BY SELECTIVE FORWARDING

by

Balaji ashok Sathyanarayanan

Bachelor of Engineering, Computer Science
University of Madras, India
2004

A thesis submitted in partial fulfillment
of the requirements for the

Master of Science Degree in Computer Science
School of Computer Science
Howard R. Hughes College of Engineering

Graduate College
University of Nevada, Las Vegas
May 2008

UMI Number: 1456371

INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

UMI®

UMI Microform 1456371
Copyright 2008 by ProQuest LLC.
All rights reserved. This microform edition is protected against unauthorized copying under Title 17, United States Code.

ProQuest LLC
789 E. Eisenhower Parkway
PO Box 1346
Ann Arbor, MI 48106-1346

UNLV
UNIVERSITY OF NEVADA LAS VEGAS

Thesis Approval
The Graduate College
University of Nevada, Las Vegas

DECEMBER 17TH, 2007

The Thesis prepared by

BALAJI ASHOK SATHYANARAYANAN

Entitled

EFFICIENT BROADCASTING BY SELECTIVE FORWARDING

is approved in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

Ajoy Kumar Datta

Examination Committee Chair

Dean of the Graduate College

John T. Minor

Examination Committee Member

Yoohwan Kim

Examination Committee Member

Graduate College Faculty Representative

1017-53

ABSTRACT

Efficient Broadcasting by Selective Forwarding

By

Balaji ahsok Sathyanarayanan

Dr. Ajoy k Datta, Examination Committee Chair
Professor of Computer Science
University of Nevada, Las Vegas

A major challenge faced in MANETs is locating the devices for communication, especially with high node mobility and sparse node density. Present solutions provided by the ad hoc routing protocols range from flooding [1] the entire network with route requests, to deploying a separate location management scheme to maintain a device location database. Many applications as well as various unicast routing protocols such as Dynamic Source Routing (DSR), Ad Hoc On Demand Distance Vector (AODV), Zone Routing Protocol (ZRP), and Location Aided Routing (LAR) use broadcasting or a derivation of it. Flooding, which is often deployed to achieve the above objective, is expensive in terms of overhead and wastes valuable resources such as bandwidth and power. We propose to develop a strategy to reduce the redundant transmission of packets in normal flooding used in broadcasting, and we describe strategies for choosing only an optimal set of nodes for rebroadcast in a grid network. The main contribution is to reduce the redundant transmission of packets and to forward packets with minimum

transmissions. To determine the minimal set of nodes for efficient transmission, we propose a new algorithm, Efficient Broadcasting by Selective Forwarding, using a distance-based approach.

The distance-based approach [2] is implemented for broadcast and rebroadcast to a set of nodes with the help of a threshold value [3] that is taken to be *threshold* = *n* transmission-radius*, where n is a real number. This reduces the number of redundant transmission. This threshold value can be tuned to show performance enhancement.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

I would like to express sincere appreciation to my thesis advisor, Dr. Ajoy K. Datta, for his guidance, insight, and support for this research. His trust and confidence in my abilities have truly encouraged me throughout my graduate study. I am also grateful to Dr.Doina Bein for helping and supporting me in completion of the thesis. I would also like to thank Dr. John Minor, Dr. Yoohwan Kim, and Dr.Venkatesan Muthukumar, for their participation on my committee.

My special gratitude goes to my family. I would like to dedicate this thesis to them for their understanding, motivation, and patience. I am thankful to all faculty members and friends who made my stay at the University of Nevada, Las Vegas a memorable and valuable experience.

CHAPTER 1

INTRODUCTION

A "mobile ad hoc network" (MANET) is an autonomous system of mobile routers (and associated nodes) connected by wireless links--the union of which forms an arbitrary graph. The routers are free to move randomly and organize themselves arbitrarily; thus, the network's wireless topology may change rapidly and unpredictably. Such a network may operate in a standalone fashion or may be connected to the larger internet. Broadcasting is the process in which one node sends a packet to all other nodes in the network. A major challenge faced in MANETs is locating the devices for communication, especially with high node mobility and sparse node density. Present solutions provided by the ad hoc routing protocols range from flooding [1] the entire network with route requests, to deploying a separate location management scheme to maintain a device location database. Many applications as well as various unicast routing protocols, such as Dynamic Source Routing (DSR), Ad Hoc on Demand Distance Vector (AODV), Zone Routing Protocol (ZRP), and Location Aided Routing (LAR), use broadcasting or a derivation of it. The principal use of flooding in these protocols is for location discovery and for establishing routes. A straightforward approach for broadcasting is *blind flooding*, in which each node will be required to rebroadcast the packet whenever it receives the packet for the first time. Blind flooding will generate many redundant transmissions, which may cause a more serious *broadcast storm problem*

[4]. Given the expensive and limited nature of wireless resources such as bandwidth and battery power, minimizing the control message overhead for route discovery is a high priority in protocol design. Recently, a number of research groups have proposed more efficient broadcasting techniques. Centralized broadcasting schemes are presented in [5-7]. Algorithms in [8-13] utilize neighborhood information to reduce redundant messages.

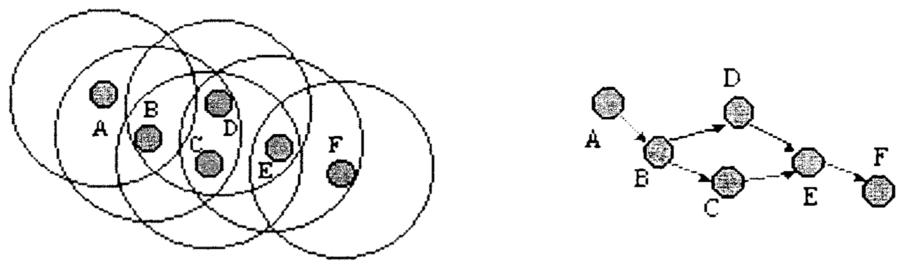


Figure 1 Ad Hoc Networks

1.1 Contribution

In this thesis we present a new protocol "Efficient Broadcasting by selective Forwarding" (EBSF) to minimize the number of transmissions/retransmissions needed for broadcasting, where only optimal nodes in the network do the retransmissions/transmission with a distance-based threshold factor. We have simulated the system with a network simulator called Glomosim. Glomosim is a library based sequential and parallel simulator for wireless networks. The library of Glomosim has been built based on the parallel discrete-event simulation capability provided by Parsec, the compiler used in Glomosim. With the simulation we have shown the performance variation of Effective Broadcasting by Selective Forwarding with different threshold factors.

1.2 Outline of the thesis

In Chapter 2 we present an overview of broadcasting, flooding, other schemes used other than flooding, and discuss why flooding is not efficient for broadcasting in MANETs. Chapter 3 includes the problem definition of Efficient Broadcasting by Selective Forwarding (ESBF), the advantages of ESBF, and the Efficient Broadcasting by Selective Forwarding protocol. Chapter 4 includes the tools used for the simulation, Glomosim, the basic structure of Glomosim, the implementation details, and the PARSEC environment which is the programming language for Glomosim. Discussion of simulation results and the performance analysis are included in Chapter 5. Finally, we summarize our research, and present further concepts for future research in Chapter 6.

CHAPTER 2

TECHNIQUES USED IN AD-HOC ROUTING PROTOCOLS

2.1 Broadcasting

Broadcasting is a communication paradigm that allows sending data packets from source to multiple receivers. In one-to-all model transmission, each node can reach all nodes that are within the radius distance from it, while in the one-to-one model, each transmission is directed towards only one neighbor. There are two different types of wireless networks:

1. The easiest network topology is where each node is able to reach all the other nodes with a traditional radio relay system with a big range. There is no use of routing protocols with this kind of network because all nodes "can see" the others.

2. The second kind uses also the radio relay system but each node has a smaller range, therefore one node has to use neighboring nodes to reach another node that is not within its transmission range. Then, the intermediate nodes are the routers. In our thesis we take into consideration the second kind of wireless network where each node has a transmission range.

Broadcasting in wireless ad-hoc networks is a critical mechanism for applications such as information diffusion, wireless networks and also for maintaining consistent global network information. Broadcasting or a derivation of it is often necessary in MANET routing protocols such as Dynamic source routing (DSR), Ad Hoc On Demand

Distance Vector (AODV), Location Aided Routing (LAR), and Zone Routing Protocol (ZRP) for establishing routes. Currently all these protocols rely on a simplistic form of broadcasting called *flooding*.

2.2 Flooding

Flooding is a process in which each node (or all nodes in a localized area) retransmits each received unique packet exactly one time. In flooding every incoming packet is sent out on every outgoing line except the one it arrived on. Whenever a device connected to the LAN (Local Area Network) switch, sends a packet to an address that is not in the LAN switch's table or whenever the device sends a broadcast or multicast packet the switch sends the packet out to all ports. This is referred to as flooding. Routers forward packets to all ports except the ingress port.

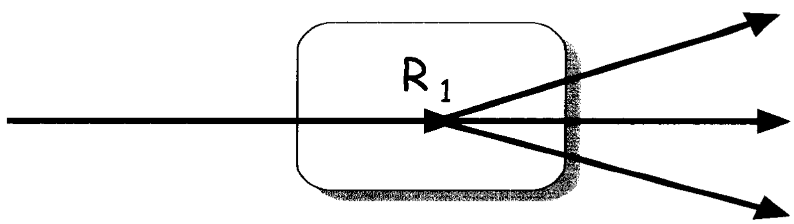


Figure 2 Typical illustration of flooding

Advantages of Flooding:

* Simple.

* Every destination in the network is reachable.

Limitations of Flooding:

* Some routers receive a packet multiple times.

* Packets can go round in loops forever.

* Inefficient usage of Bandwidth.

* More power consumption.

* Leads to transmission of redundant packets.

* Broadcast storm problem: For dense networks, flooding is very inefficient and causes significant contention and collisions.

2.3 Schemes used other than flooding

Several schemes have been developed for improving the efficiency of broadcasting by reducing the number of broadcasts. Some of them are described below.

2.3.1 Probabilistic scheme

The probabilistic scheme [13, 20] is similar to flooding, except that nodes only rebroadcast with predetermined probability. Some nodes do not rebroadcast and this saves node and network resources without having delivery effectiveness. In sparse networks, there is much less shared coverage. Thus, nodes won't receive all the broadcast packets with the probabilistic scheme unless the probability parameter is high. When the probability is 100%, this scheme is identical to flooding.

2.3.2 Counter-based scheme

There is an inverse relationship between the number of times a packet is received at a node and the probability of that node being able to reach additional area on a broadcast. This result is the basis of their counter-based scheme [4, 13, 20]. Upon reception of a previously unseen packet, the node initiates a counter with a value one and sets a RAD (Random Assessment Delay). During the RAD, the counter is incremented by one for each redundant packet received. If the counter is less than a threshold value when the RAD expires, then the packet is rebroadcast, otherwise the packet is simply dropped.

Advantages of probabilistic and counter-based schemes [4]:

* Simplicity

* Inherent adaptability to local topologies

Disadvantages:

* Delivery is not guaranteed to all nodes even if ideal MAC is provided. i.e., they are not reliable.

2.3.3 Area Based decisions

Area based methods [20] only consider the coverage area of transmission. They don't consider whether nodes exist within that area.

Two coverage-area based methods are:

1. Distance-Based scheme [13].

2. Location Based scheme [13].

In the above two schemes a node decides whether to rebroadcast purely based on its own information.

In this thesis we have developed a new approach to the distance-based scheme [13] for broadcasting, called Efficient Broadcasting by Selective Forwarding (ESBF), where broadcasting is done by a particular node which privileged to broadcast based on the distance (threshold distance). If the threshold distance is set to zero the network broadcasting will be similar to flooding.

2.3.4 Why not flooding in broadcasting

A uniform distribution of nodes is an effective way to organize a network consisting of a large number of nodes. Flooding provides important control and route establishment functionality for a number of unicast and multicast protocols in ad hoc networks. Considering its wide use as a building layer for other network layer protocols, the flooding methodology should deliver a packet from one node to all other network nodes using as few messages as possible. Also network wide broadcasting is an energy intensive function. Also in mobile ad hoc networks, it is often necessary to broadcast control information to all the constituent nodes in the network. Flooding, which is often deployed to achieve the above objective, is expensive in terms of overhead and wastes valuable resources such as bandwidth and power. An improvement to flooding is to choose only a subset of nodes to rebroadcast and thus reduce data transmissions.

CHAPTER 3

EFFICIENT BROADCASTING BY SELECTIVE FORWARDING

3.1 Efficient Broadcasting by Selective Forwarding

Broadcasting is a protocol which allows a process to send one message out, and allows all of the other processes to read that single message. In this thesis we devise an Efficient Broadcasting by Selective Forwarding algorithm for choosing only an optimal set of nodes for rebroadcast. The main contribution is to reduce the redundant transmission of packets and to reduce the redundant transmissions to broadcast the messages throughout the network.

We develop a strategy to reduce the redundant transmission of packets in normal flooding used in broadcasting. We have devised a mechanism where an optimal selection is done to determine the minimal set of nodes for rebroadcast in mobile ad hoc networks. To determine the minimal set of nodes we employ the distance-based approach by which the data transmitted is reduced considerably to a greater extent where all data is received by all nodes in the network, by using Efficient Broadcasting by Selective Forwarding.

The Distance-based approach is implemented with the help of a threshold value that is taken to be *threshold* = *N* * *transmission radius*, where N has values $0 < N < 1$. This threshold value can be tuned as per requirements that show the performance enhancement. The Higher the threshold, the lesser is the number of packets transmitted by each node in the network and vice versa. Thus this threshold value can be increased and decreased to show the performance variation. If threshold is set to 0, then all the packets are transmitted like in flooding. Hence no enhancement is seen.

We have also developed a strategy to determine the optimal set of nodes by using the transmission range of a node with the help of which the distance-based approach is implemented. We have also implemented a new data structure called as message cache that can be used for checking duplicate messages to reduce redundant messages being delivered.

All the above said strategies are developed with a new protocol, EBSF, which is embedded in the network layer of the simulator Glomosim.

3.2 Advantages of EBSF

* Selection of optimal number of nodes to retransmit.
* Effective utilization of bandwidth.
* Minimizing the number of unnecessary transmissions and therefore reducing redundant packets.
* Minimizing the power consumption.

The main contribution of our research is to reduce the redundant transmission of packets in normal flooding used in broadcasting. We have devised a mechanism where an optimal selection is done to determine the minimal set of nodes for re-broadcasting in mobile ad-hoc networks by Efficient Broadcasting by Selective Forwarding.

Initial Assumptions:

1. Uniform transmission power for all the nodes (constant radius).
2. The node placement format is a grid.

3. The nodes are almost uniformly distributed in the network.

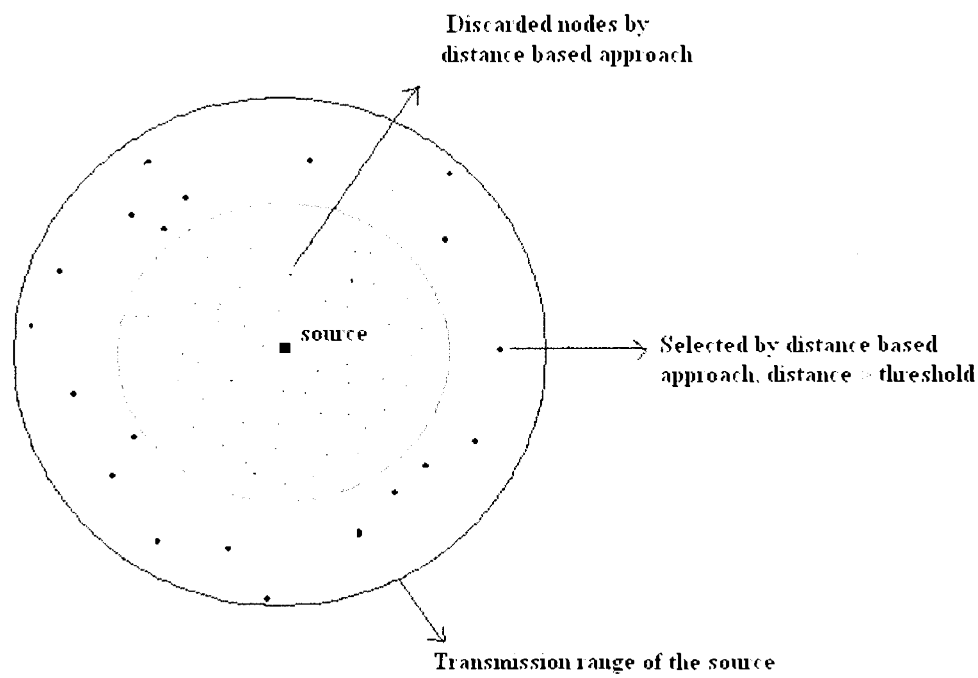


Figure 3 Efficient Broadcasting by Selective Forwarding

3.3 Efficient Broadcasting by Selective Forwarding protocol (ESBF)

Efficient Broadcasting by Selective Forwarding is a distance-based approach for broadcasting. In flooding, location discovery is achieved by allowing all the nodes that receive the message to retransmit. We restrict the retransmission by all the nodes with a distance factor which is the threshold. In a network not all the nodes have to transmit/retransmit. Instead a few strategically selected nodes could be selected for retransmission which would reduce the redundant transmissions.

The OSI 7-layer model has clear characteristics at each layer. Layers 7 through 4 deal with end to end communications between data source and destinations, while layers 3 to

1 deal with communications between network devices. On the other hand, the seven layers of the OSI model can be divided into two groups: upper layers (layers 7, 6 & 5) and lower layers (layers 4, 3, 2, 1). The upper layers of the OSI model deal with application issues and generally are implemented only in software. The highest layer, the application layer, is closest to the end user. The lower layers of the OSI model handle data transport issues, of which the network layer is of importance to us.

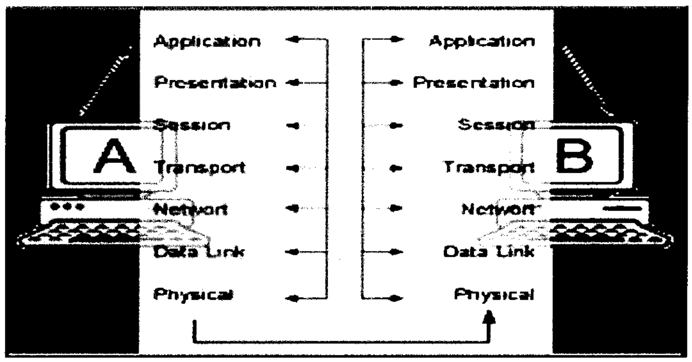


Figure 4 OSI layers

Layer 3: Network Layer

- Determines how data are transferred between network devices
- Routes packets according to unique network device addresses
- Provides flow and congestion control to prevent network resource depletion

When a packet is to be transmitted from source node to a destination node the network layer is responsible for end to end (source to destination) packet delivery, whereas the data link layer is responsible for node to node (hop to hop) packet delivery. When a node receives a packet the MAC layer of the data link layer sends the packet to the network layer for transmission.

We are concerned with the network layer as Efficient Broadcasting by Selective Forwarding is implemented in the network layer.

When a node A broadcasts P packets to other nodes in the network, each node has a message cache (MSGCACHE) which stores the information about messages the node has received. When a node receives a packet, it checks with the Message cache if the packet has already been transmitted to the node. If the packet has already been received, it discards the message. If not received it will be forwarded to the network layer for transmission. Thus it checks for redundant data and discards the redundant packets. The message cache is implemented as a linked list data structure.

When a packet is forwarded to the network layer for transmission, the IP header in the Internet Protocol has the source node coordinates and the destination node coordinates which have the location of the source and destination node. We calculate the distance between the source and destination nodes. In flooding whenever a message is received for transmission in the network layer, it broadcasts the packet to all the nodes in its transmission range once it receives it from the MAC layer. Whereas in Efficient Broadcasting by Selective Forwarding the node which has to transmit the message will calculate the distance of nodes which are equal to or more than the threshold distance. Nodes which are at a distance greater than or equal to the threshold are allowed to retransmit the messages. The nodes upon receiving a message, stores the message information in the message cache and looks for nodes at a distance equal to or greater than the threshold distance. It is more likely that there may be a node that has received a message which will fall in the threshold distance of another node where it would check if the message information is in the MSGCACHE. Once a node receives a redundant message it will discard the message.

In Efficient Broadcasting by Selective Forwarding (ESBF) when a packet reaches the network layer of a node for broadcasting, the following computation is performed. Based on the information about the source and destination in the Internet protocol, we calculate the distance between the source node and the destination node. When the distance between the source node and destination node is less than the threshold distance, the node discards the message; if the distance between the source node and destination node is greater than the threshold distance, it will broadcast the message to the nodes at a distance equal to or greater than the threshold distance. The threshold distance is defined as N*transmission range of a node, where N is a real number between 0 and 1. By limiting the transmission to the nodes at the threshold distance from a node, we reduce the number of transmissions by allowing only a few nodes in the transmission range of the node to retransmit the message. We can tune the value of N to show performance variations. If N is set to zero then the broadcasting is done as in blind flooding. The Higher the threshold, the lesser is the number of packets transmitted and vice versa. Thus this threshold value can be increased and decreased to show the performance variation.

Every node on receiving a message will check for redundancy of the message with the MSGCACHE and send it to the network layer for broadcast. The network layer will check for the threshold distance between the source node and destination nodes and decide whether to broadcast or not to broadcast based on the distance between the source node and destination node. This is repeated until all the nodes in the network receive all the packets originated from the source node.

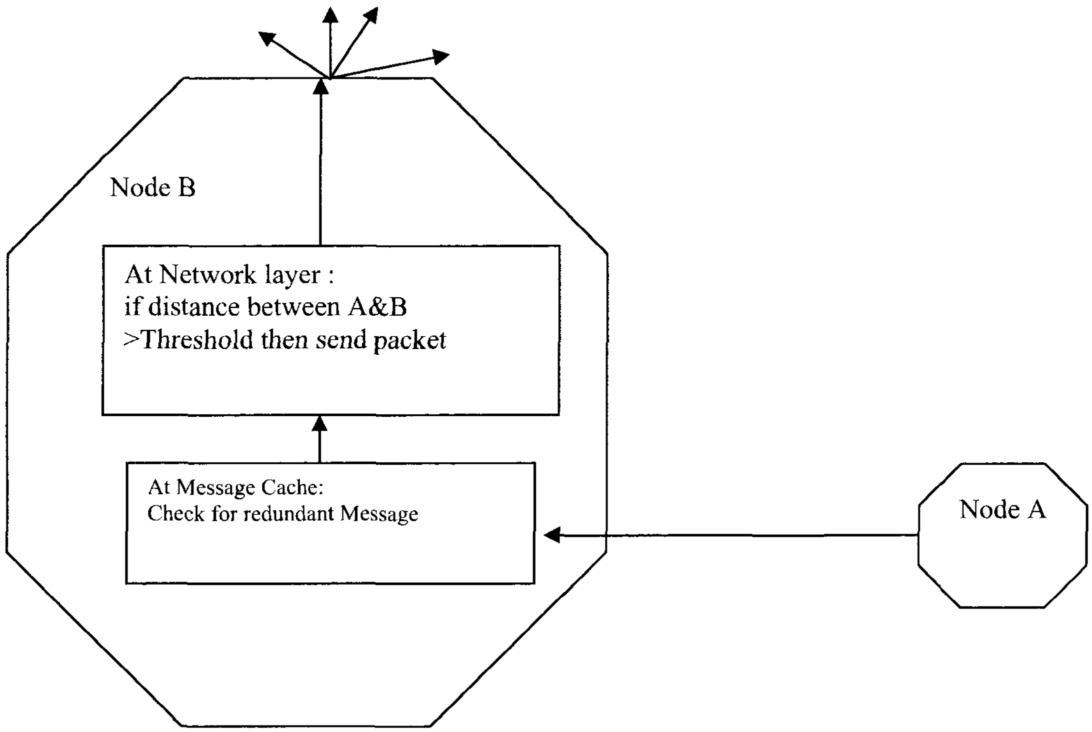


Figure 5 Broadcasting protocol

3.4 Concept of EBSF

Assumptions are made that each node knows its location; every node has the same transmission range, and the distance from each node in the network. When a node receives a broad cast packet it discards the packet if the following conditions are not satisfied.

Condition 1: Check the message cache to see if it has already received the message; If the message is received then discard the message.

Condition 2: If the distance to the source node is lesser than the threshold distance.

The node accepts a packet and sends it further if the following conditions are satisfied:

Condition 2.1: If the message is not in the message cache,

Condition 2.2: If the distance to the source is greater than or equal to the threshold distance.

3.5 Further enhancement of EBSF

When we consider a node, we can separate its neighbors into two types, *geographical neighbors* and *communicational neighbors*.

Geographical neighbors of a node in a grid network are the nodes that are located around a node at a distance of one grid unit from the node. A node at the corner of a grid will have two geographical neighbors, and a node at the border of the grid will have three geographical neighbors. All other nodes have four geographical neighbors.

Communicational neighbors are the nodes that are within the transmission range of a node excluding the geographical nodes.

We have enhanced EBSF to exclude the geographical neighbors of a node which are at threshold distance and have thereby decreased the number of transmissions further. For example if node A is at a distance greater than or equal to the threshold distance of node B which is the sender, then more likely the geographical nodes of node A are also at the threshold distance from the node B. In this case node A and the geographical neighbors of A are allowed to broadcast the message received from node B in EBSF. Where node A and its geographical neighbors may have common nodes which are at a threshold distance from them and these nodes will receive the same messages from node A and its geographical neighbors, which would lead to redundant transmissions. Hence with the enhanced EBSF algorithm we restrict the geographical neighbors of a node to not broadcast and thereby reduce redundant transmissions, which is evident from the simulation results.

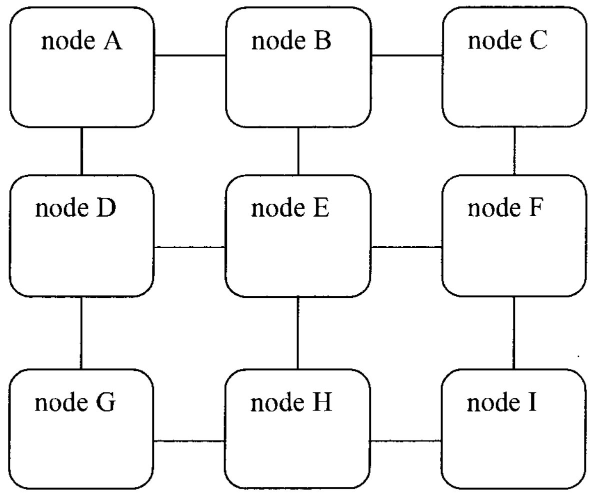


Figure 6 Network showing geographical neighbors

From the above figure we can see node E's geographical neighbors are node B, D, F, and H. Node G which is a corner node has two geographical neighbors ,nodes D and H, and node H which is a border node has three neighboring nodes D, E, I. The communicational neighbors are the nodes that are within the transmission range of a node excluding the geographical neighbors.

CHAPTER 4

TOOL USED FOR THE THESIS

4.1 Glomosim

The tool we have used for the simulation is Glomosim (Global mobile system simulator)

Glomosim is a scalable simulation environment for wireless and wired systems. It was designed using the parallel discrete-event simulation capability provided by Parsec. Glomosim currently supports protocols for a purely wireless network. The Parsec compiler is similar to a 'C' compiler with some added functions. The .pc files are written in parsec language. Any user who is familiar with 'C' can write a parsec program. It is not compulsory for the user to be familiar with the statements exclusive to parsec (e.g. entity, send, receive statements).

The simulation statistics will be stored in the bin directory of Glomosim as a glomo.stat file. Glomosim is a very useful, versatile network simulator. The layers in Glomosim are shown in Table 1.

Table 1. Layers of Glomosim

LAYERS	PROTOCOLS
Mobility	Random waypoint, Random drunken, Trace based
Radio Propagation	Two ray and Free space
Radio Model	Noise Accumulating
Packet Reception Models	SNR bounded, BER based with BPSK/QPSK modulation
Data Link (MAC)	CSMA, IEEE 802.11 and MACA
Network (Routing)	IP with AODV, Bellman-Ford, DSR, Fisheye, LAR scheme 1, ODMRP
Transport	TCP and UDP
Application	CBR, FTP, HTTP and Telnet

4.2 Structure of Glomosim

Glomosim contains the following directories:

application contains code for the application layer

bin for executable and input/output files

doc contains the documentation

include contains common include files

java gui contains the visual tool

Mac contains the code for the Mac layer

main contains the basic framework design

network contains the code for the network layer

radio contains the code for the radio layer

scenarios contains some example scenarios

tcplib contains libraries for TCP

transport contains the code for the transport layer

To accept EBSF as a valid routing protocol we have programmed and modified the following files in Glomosim .The files that were modified are:

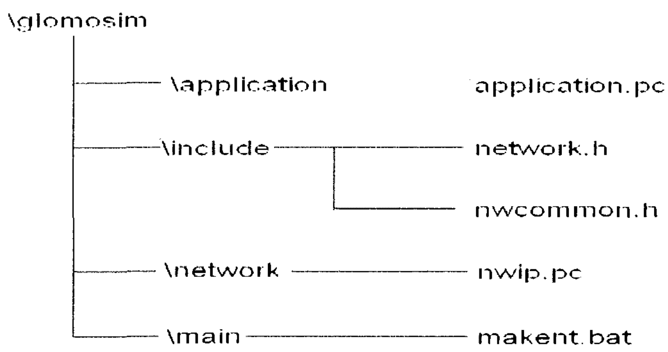


Figure 7 Modified files in Glomosim

All routing decisions are taken in the network layer, so we embed our protocol EBSF in the network layer. For our protocol EBSF to work in the network layer of Glomosim we have modified and included ESBF.H and EBSF.PC in the network directory of Glomosim.

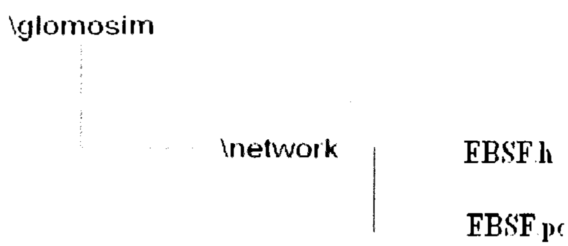


Figure 8 Files programmed for EBSF

The program embedded in EBSF.PC contains the code for threshold determination and the code to determine whether to broadcast or to discard a message.

4.3 Basic Design of the System

4.3.1 Module Diagram

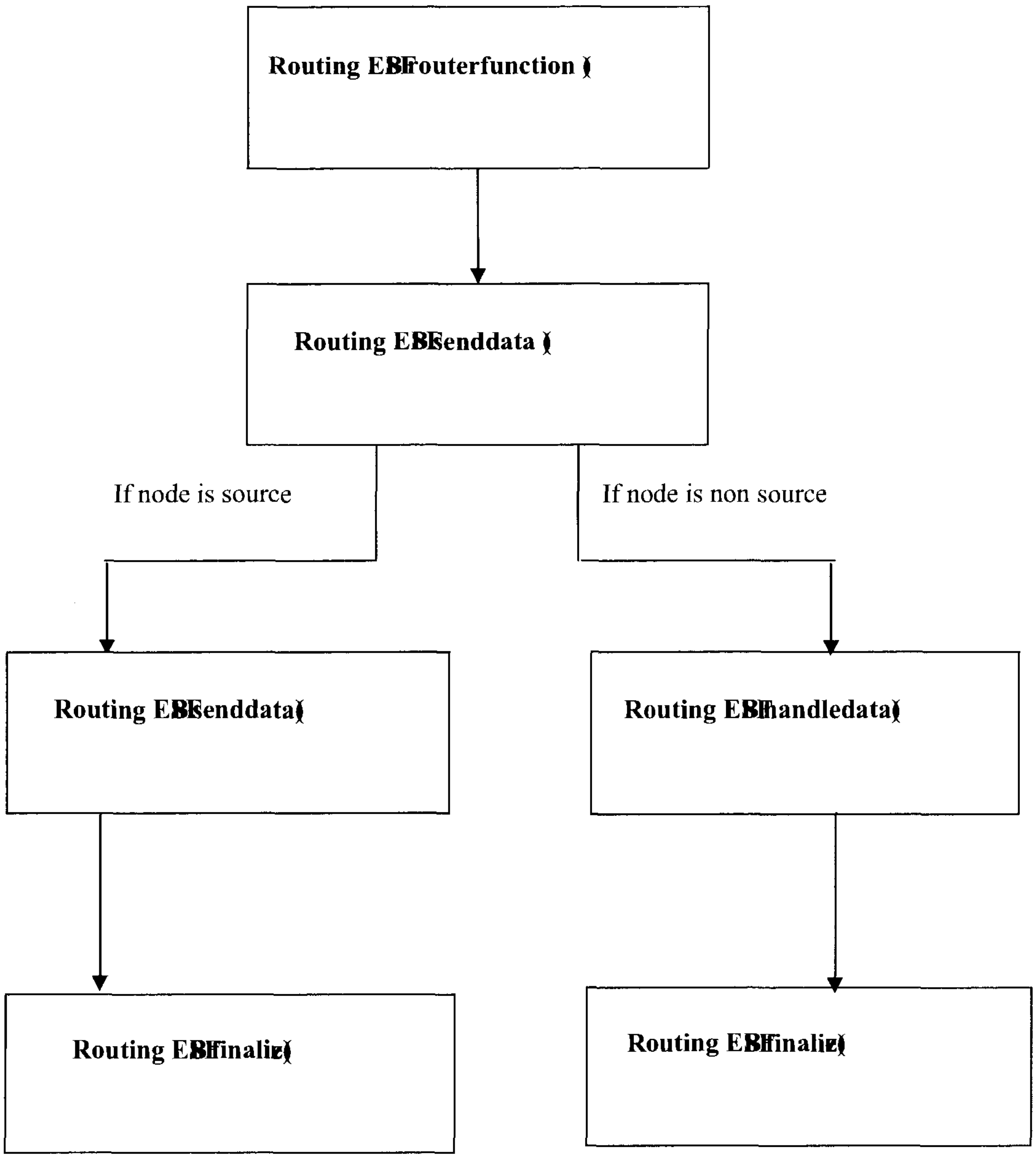


Figure 9 Module diagram of EBSF

4.4 Implementation Details

4.4.1 Implementation of EBSF

We have implemented our protocol EBSF (Efficient Broadcasting by Selective Forwarding) in the network layer of the Glomosim. This protocol will be embedded in each node of the network. Whenever a packet enters the network layer, the packet will be handled by this protocol EBSF.

Initialization Function:

Whenever this protocol is used, the initialization function RoutingEBSFInit() is called that defines a structure called GlomoRoutingEBSF and memory is allocated for it. Also we initialize the stats (statistics to be printed to determine the enhancements), sequence table for the node, and message cache. Top value is initialized to 0.

Router Function:

A routing function RoutingEBSFRouterFunction () determines the routing action to be taken and handles the packet accordingly if it is from UDP or MAC. a) If the packet is from UDP, then the node is the source node and we send the data i.e., the function RoutingEBSFSendData () is called. b) If data comes from MAC layer, we handle the data and then the decision is made as whether to send it to UDP or drop the packet. This decision is made in the RoutingEBSFHandleData() function.

Finalize Function:

The finalize function initializes the statistics part of the protocol. When this function is called it collects the statistics from the glomo.stat file in the BIN directory and formats the statistics such that number of data transmitted, number of data originated, and number of data received for each node are printed.

Data Handling Function:

The function RoutingEBSFHandleData( ) is called whenever the node receives the packet from MAC layer. This function checks with the message cache and it decides whether to transmit or discard the packet.

Here all the nodes that are within the transmission range will receive the packets, and only the nodes that are greater than the threshold distance and lesser than the transmission range will transmit the packets. If the node is at a distance which is lesser than a threshold value from a transmitted node, then the packet is discarded.

In our code (Efficient Broadcasting by Selective Forwarding):

1. Duplicate message checking is done;

2. If the packet is a duplicate i.e., if it has already been received by the node (lookupmessagecache), then the packet is discarded. Otherwise, the distance from the sender is calculated;

3. If the distance value is less than threshold i.e., N*transmission range of node, then the packet is discarded. The Higher the threshold, the smaller is the number of packets transmitted, and vice versa. Thus the threshold value is increased and decreased to show the performance variation. If threshold is set to 0, then all the packets are transmitted. Hence no enhancement is seen;

4. Everytime, the node's location and node coordinates are updated to the previous location and coordinates.

4.4.2 Implementation of Message Cache

The message cache is implemented as a linked list into which is inserted a message, i.e. source address and sequence number, whenever a new message arrives at a node. The function LookupMessageCache searches whether the message is there in the cache, by its sequence number, and function InsertMessageCache inserts a message into the cache if it is not already present there.

4.5 Internals of Glomosim

- The structure contains information about each particular node. It can be found in ./include/api.h

  struct glomo_node_str { … }

- Generic message sent by any layer can be found in ./include/message.h

  struct message { …. }

Other packet functions are GLOMO_MsgAlloc, GLOMO_MsgAddHeader, GLOMO_MsgRemoveHeader, GLOMO_MsgPacketAlloc.

4.6 The Parsec Programming Environment

The PARSEC compiler, called ***pcc***, accepts all the options supported by the C compiler, and also supports separate compilation. C programs (files with .c suffix) and object files (files with .o suffix) can also be compiled and linked with PARSEC programs. PARSEC programs are usually given a .pc extension. PARSEC supports separate compilation of entities.

PCC_DIRECTORY - installation directory for PARSEC.

PCC_CC - C compiler for PARSEC to use.

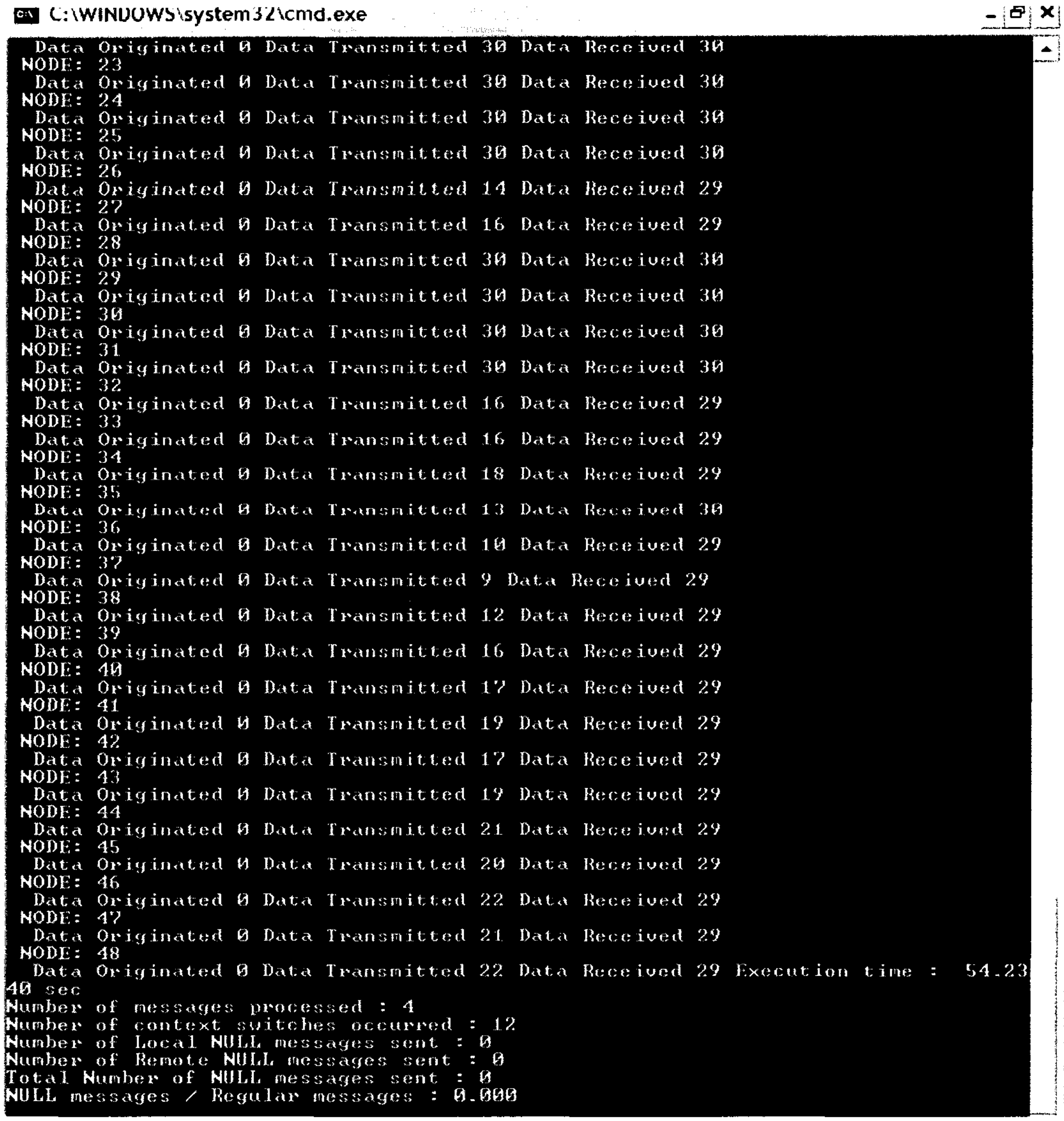
PCC_LINKER - linker for PARSEC to use.

CHAPTER 5

SIMULATION AND RESULTS

5.1 Result of PARSEC

The following window shows the PARSEC compilation window where the statistics are shown for each node, i.e. about the number of data originated at the node, number of data transmitted, and number of data received by each node.



```
C:\WINDOWS\system32\cmd.exe
 Data Originated 0 Data Transmitted 30 Data Received 30
NODE: 23
 Data Originated 0 Data Transmitted 30 Data Received 30
NODE: 24
 Data Originated 0 Data Transmitted 30 Data Received 30
NODE: 25
 Data Originated 0 Data Transmitted 30 Data Received 30
NODE: 26
 Data Originated 0 Data Transmitted 14 Data Received 29
NODE: 27
 Data Originated 0 Data Transmitted 16 Data Received 29
NODE: 28
 Data Originated 0 Data Transmitted 30 Data Received 30
NODE: 29
 Data Originated 0 Data Transmitted 30 Data Received 30
NODE: 30
 Data Originated 0 Data Transmitted 30 Data Received 30
NODE: 31
 Data Originated 0 Data Transmitted 30 Data Received 30
NODE: 32
 Data Originated 0 Data Transmitted 16 Data Received 29
NODE: 33
 Data Originated 0 Data Transmitted 16 Data Received 29
NODE: 34
 Data Originated 0 Data Transmitted 18 Data Received 29
NODE: 35
 Data Originated 0 Data Transmitted 13 Data Received 30
NODE: 36
 Data Originated 0 Data Transmitted 10 Data Received 29
NODE: 37
 Data Originated 0 Data Transmitted 9 Data Received 29
NODE: 38
 Data Originated 0 Data Transmitted 12 Data Received 29
NODE: 39
 Data Originated 0 Data Transmitted 16 Data Received 29
NODE: 40
 Data Originated 0 Data Transmitted 17 Data Received 29
NODE: 41
 Data Originated 0 Data Transmitted 19 Data Received 29
NODE: 42
 Data Originated 0 Data Transmitted 17 Data Received 29
NODE: 43
 Data Originated 0 Data Transmitted 19 Data Received 29
NODE: 44
 Data Originated 0 Data Transmitted 21 Data Received 29
NODE: 45
 Data Originated 0 Data Transmitted 20 Data Received 29
NODE: 46
 Data Originated 0 Data Transmitted 22 Data Received 29
NODE: 47
 Data Originated 0 Data Transmitted 21 Data Received 29
NODE: 48
 Data Originated 0 Data Transmitted 22 Data Received 29 Execution time :  54.23
40 sec
Number of messages processed : 4
Number of context switches occurred : 12
Number of Local NULL messages sent : 0
Number of Remote NULL messages sent : 0
Total Number of NULL messages sent : 0
NULL messages / Regular messages : 0.000
```

Figure 10 Statistics of EBSF

5.2 Sample of statistics generated by Glomosim

The following is a part of the statistics generated by Glomosim for a network of 49 nodes and threshold factor N=0.6.

Node 0 (0.00, 0.00, 0.00).
Node 1 (0.00, 25.00, 0.00).
Node 2 (0.00, 50.00, 0.00).
Node 3 (0.00, 75.00, 0.00).
Node 4 (0.00, 100.00, 0.00).

Node 5 (0.00, 125.00, 0.00).

.

.

.

.

Node 44 (150.00, 50.00, 0.00).
Node 45 (150.00, 75.00, 0.00).
Node 46 (150.00, 100.00, 0.00).
Node 47 (150.00, 125.00, 0.00).
Node 48 (150.00, 150.00, 0.00).

Current Sim Time[s] = 0.000000000 Real Time[s] = 0 Completed 0%
Current Sim Time[s] = 5.000000000 Real Time[s] = 0 Completed 1%

SOURCE 1
Current Sim Time[s] = 5.000001000 Real Time[s] = 0 Completed 2%
Current Sim Time[s] = 5.000046000 Real Time[s] = 0 Completed 3%
Current Sim Time[s] = 5.000051000 Real Time[s] = 0 Completed 4%
Current Sim Time[s] = 5.000051083 Real Time[s] = 0 Completed 5%
Current Sim Time[s] = 5.000051118 Real Time[s] = 0 Completed 6%
Current Sim Time[s] = 5.000051167 Real Time[s] = 0 Completed 7%
Current Sim Time[s] = 5.000051186 Real Time[s] = 0 Completed 8%
Current Sim Time[s] = 5.000051236 Real Time[s] = 0 Completed 9%
Current Sim Time[s] = 5.000051250 Real Time[s] = 0 Completed 10%
Current Sim Time[s] = 5.000051264 Real Time[s] = 0 Completed 11%
Current Sim Time[s] = 5.000051300 Real Time[s] = 0 Completed 12%
Current Sim Time[s] = 5.000051333 Real Time[s] = 0 Completed 13%
Current Sim Time[s] = 5.000051344 Real Time[s] = 0 Completed 14%
Current Sim Time[s] = 5.000051354 Real Time[s] = 0 Completed 15%
Current Sim Time[s] = 5.000051373 Real Time[s] = 0 Completed 16%
Current Sim Time[s] = 5.000051417 Real Time[s] = 0 Completed 17%
Current Sim Time[s] = 5.000051425 Real Time[s] = 0 Completed 18%
Current Sim Time[s] = 5.000051449 Real Time[s] = 0 Completed 19%
Current Sim Time[s] = 5.000051471 Real Time[s] = 0 Completed 20%

NODE:0 Received from node 1 with location (0.000000 ,25.000000 ,0.000000)
Threshold Distance is: 70.933200
Distance from the previous Node: 25.000000
Node 0 Message Discarded(distance < threshold)

NODE:2 Received from node 1 with location (0.000000 ,25.000000 ,0.000000)
Threshold Distance is: 70.933200
Distance from the previous Node: 25.000000
Node 2 Message Discarded(distance < threshold)

NODE:8 Received from node 1 with location (0.000000 ,25.000000 ,0.000000)

Threshold Distance is: 70.933200
Distance from the previous Node: 25.000000
Node 8 Message Discarded(distance < threshold)

NODE:7 Received from node 1 with location (0.000000 ,25.000000 ,0.000000)
Threshold Distance is: 70.933200
Distance from the previous Node: 35.355339
Node 7 Message Discarded(distance < threshold)

NODE:9 Received from node 1 with location (0.000000 ,25.000000 ,0.000000)
Threshold Distance is: 70.933200
Distance from the previous Node: 35.355339
Node 9 Message Discarded(distance < threshold)

NODE:3 Received from node 1 with location (0.000000 ,25.000000 ,0.000000)
Threshold Distance is: 70.933200
Distance from the previous Node: 50.000000
Node 3 Message Discarded(distance < threshold)

NODE:15 Received from node 1 with location (0.000000 ,25.000000 ,0.000000)
Threshold Distance is: 70.933200
Distance from the previous Node: 50.000000
Node 15 Message Discarded(distance < threshold)

NODE:10 Received from node 1 with location (0.000000 ,25.000000 ,0.000000)
Threshold Distance is: 70.933200
Distance from the previous Node: 55.901699
Node 10 Message Discarded(distance < threshold)

NODE:14 Received from node 1 with location (0.000000 ,25.000000 ,0.000000)
Threshold Distance is: 70.933200
Distance from the previous Node: 55.901699
Node 14 Message Discarded(distance < threshold)

NODE:16 Received from node 1 with location (0.000000 ,25.000000 ,0.000000)
Threshold Distance is: 70.933200
Distance from the previous Node: 55.901699
Node 16 Message Discarded(distance < threshold)

NODE:17 Received from node 1 with location (0.000000 ,25.000000 ,0.000000)
Threshold Distance is: 70.933200
Distance from the previous Node: 70.710678
Node 17 Message Discarded(distance < threshold)

NODE:4 Received from node 1 with location (0.000000 ,25.000000 ,0.000000)
Threshold Distance is: 70.933200

Distance from the previous Node: 75.000000
The Node 4 Sends message

NODE:22 Received from node 1 with location (0.000000 ,25.000000 ,0.000000)
Threshold Distance is: 70.933200
Distance from the previous Node: 75.000000
The Node 22 Sends message

NODE:11 Received from node 1 with location (0.000000 ,25.000000 ,0.000000)
Threshold Distance is: 70.933200
Distance from the previous Node: 79.056942
The Node 11 Sends message

NODE:21 Received from node 1 with location (0.000000 ,25.000000 ,0.000000)
Threshold Distance is: 70.933200
Distance from the previous Node: 79.056942
The Node 21 Sends message

NODE:23 Received from node 1 with location (0.000000 ,25.000000 ,0.000000)
Threshold Distance is: 70.933200
Distance from the previous Node: 79.056942
The Node 23 Sends message

NODE:18 Received from node 1 with location (0.000000 ,25.000000 ,0.000000)
Threshold Distance is: 70.933200
Distance from the previous Node: 90.138782
The Node 18 Sends message

NODE:24 Received from node 1 with location (0.000000 ,25.000000 ,0.000000)
Threshold Distance is: 70.933200
Distance from the previous Node: 90.138782
The Node 24 Sends message

NODE:5 Received from node 1 with location (0.000000 ,25.000000 ,0.000000)
Threshold Distance is: 70.933200
Distance from the previous Node: 100.000000
The Node 5 Sends message

NODE:29 Received from node 1 with location (0.000000 ,25.000000 ,0.000000)
Threshold Distance is: 70.933200
Distance from the previous Node: 100.000000
The Node 29 Sends message

NODE:12 Received from node 1 with location (0.000000 ,25.000000 ,0.000000)
Threshold Distance is: 70.933200
Distance from the previous Node: 103.077641

The Node 12 Sends message

NODE:28 Received from node 1 with location (0.000000 ,25.000000 ,0.000000)
Threshold Distance is: 70.933200
Distance from the previous Node: 103.077641
The Node 28 Sends message

NODE:30 Received from node 1 with location (0.000000 ,25.000000 ,0.000000)
Threshold Distance is: 70.933200
Distance from the previous Node: 103.077641
The Node 30 Sends message

NODE:25 Received from node 1 with location (0.000000 ,25.000000 ,0.000000)
Threshold Distance is: 70.933200
Distance from the previous Node: 106.066017
The Node 25 Sends message
NODE:19 Received from node 1 with location (0.000000 ,25.000000 ,0.000000)
Threshold Distance is: 70.933200
Distance from the previous Node: 111.803399
The Node 19 Sends message

NODE:31 Received from node 1 with location (0.000000 ,25.000000 ,0.000000)
Threshold Distance is: 70.933200
Distance from the previous Node: 111.803399
The Node 31 Sends message

Node 15 Received Duplicate message

Node 21 Received Duplicate message

Node 23 Received Duplicate message
Node 29 Received Duplicate message

Node 14 Received Duplicate message

Node 16 Received Duplicate message

Node 28 Received Duplicate message

Node 30 Received Duplicate message

Node 8 Received Duplicate message

Node 24 Received Duplicate message

NODE:36 Received from node 22 with location (75.000000 ,25.000000 ,0.000000)

Threshold Distance is: 70.933200
Distance from the previous Node: 50.000000
Node 36 Message Discarded(distance < threshold)

Node 7 Received Duplicate message

Node 9 Received Duplicate message

Node 17 Received Duplicate message

Node 31 Received Duplicate message

NODE:35 Received from node 22 with location (75.000000 ,25.000000 ,0.000000)
Threshold Distance is: 70.933200
Distance from the previous Node: 55.901699
Node 35 Message Discarded(distance < threshold)

NODE:37 Received from node 22 with location (75.000000 ,25.000000 ,0.000000)
Threshold Distance is: 70.933200
Distance from the previous Node: 55.901699
Node 37 Message Discarded(distance < threshold)

Node 10 Received Duplicate message

NODE:38 Received from node 22 with location (75.000000 ,25.000000 ,0.000000)
Threshold Distance is: 70.933200
Distance from the previous Node: 70.710678
Node 38 Message Discarded(distance < threshold)

Node 1 Received Duplicate message

Node 25 Received Duplicate message

NODE:43 Received from node 22 with location (75.000000 ,25.000000 ,0.000000)
Threshold Distance is: 70.933200
Distance from the previous Node: 75.000000
The Node 43 Sends message

Node 0 Received Duplicate message

Node 2 Received Duplicate message

Node 18 Received Duplicate message

NODE:32 Received from node 22 with location (75.000000 ,25.000000 ,0.000000)
Threshold Distance is: 70.933200

Distance from the previous Node: 79.056942
The Node 32 Sends message

NODE:42 Received from node 22 with location (75.000000 ,25.000000 ,0.000000)
Threshold Distance is: 70.933200
Distance from the previous Node: 79.056942
The Node 42 Sends message

NODE:44 Received from node 22 with location (75.000000 ,25.000000 ,0.000000)
Threshold Distance is: 70.933200
Distance from the previous Node: 79.056942
The Node 44 Sends message

Node 3 Received Duplicate message

Node 11 Received Duplicate message

NODE:39 Received from node 22 with location (75.000000 ,25.000000 ,0.000000)
Threshold Distance is: 70.933200
Distance from the previous Node: 90.138782
The Node 39 Sends message

NODE:45 Received from node 22 with location (75.000000 ,25.000000 ,0.000000)
Threshold Distance is: 70.933200
Distance from the previous Node: 90.138782
The Node 45 Sends message

NODE:26 Received from node 22 with location (75.000000 ,25.000000 ,0.000000)
Threshold Distance is: 70.933200
Distance from the previous Node: 100.000000
The Node 26 Sends message

Node 19 Received Duplicate message

NODE:33 Received from node 22 with location (75.000000 ,25.000000 ,0.000000)
Threshold Distance is: 70.933200
Distance from the previous Node: 103.077641
The Node 33 Sends message

Node 4 Received Duplicate message

NODE:46 Received from node 22 with location (75.000000 ,25.000000 ,0.000000)
Threshold Distance is: 70.933200
Distance from the previous Node: 106.066017
The Node 46 Sends message

Node 12 Received Duplicate message

NODE:40 Received from node 22 with location (75.000000 ,25.000000 ,0.000000)
Threshold Distance is: 70.933200
Distance from the previous Node: 111.803399
The Node 40 Sends message

Node 4 Received Duplicate message

Node 10 Received Duplicate message

Node 12 Received Duplicate message

Node 18 Received Duplicate message

Node 3 Received Duplicate message

Node 5 Received Duplicate message

Node 19 Received Duplicate message

NODE:13 Received from node 11 with location (25.000000 ,100.000000 ,0.000000)
Threshold Distance is: 70.933200
Distance from the previous Node: 50.000000
Node 13 Message Discarded(distance < threshold)

NODE:6 Received from node 11 with location (25.000000 ,100.000000 ,0.000000)
Threshold Distance is: 70.933200
Distance from the previous Node: 55.901699
Node 6 Message Discarded(distance < threshold)

Node 36 Received Duplicate message

Node 42 Received Duplicate message

Node 44 Received Duplicate message

Node 35 Received Duplicate message

Node 4 Received Duplicate message

Node 6 Received Duplicate message

Node 12 Received Duplicate message

Node 11 Received Duplicate message
Node 13 Received Duplicate message
Node 3 Received Duplicate message

Node 19 Received Duplicate message

Node 10 Received Duplicate message

Node 18 Received Duplicate message

NODE:20 Received from node 5 with location (0.000000 ,125.000000 ,0.000000)
Threshold Distance is: 70.933200
Distance from the previous Node: 55.901699
Node 20 Message Discarded(distance < threshold)

Node 17 Received Duplicate message

Node 2 Received Duplicate message

Node 26 Received Duplicate message

Node 9 Received Duplicate message

Node 25 Received Duplicate message

NODE:27 Received from node 5 with location (0.000000 ,125.000000 ,0.000000)
Threshold Distance is: 70.933200
Distance from the previous Node: 79.056942
The Node 27 Sends message

Node 16 Received Duplicate message

Node 24 Received Duplicate message

Node 1 Received Duplicate message

Node 33 Received Duplicate message

Node 8 Received Duplicate message

Node 32 Received Duplicate message

NODE:34 Received from node 5 with location (0.000000 ,125.000000 ,0.000000)
Threshold Distance is: 70.933200
Distance from the previous Node: 103.077641
The Node 34 Sends message

Node 23 Received Duplicate message
Node 15 Received Duplicate message

Node 31 Received Duplicate message

Node 20 Received Duplicate message

Node 26 Received Duplicate message
Node 34 Received Duplicate message

Node 19 Received Duplicate message

Node 33 Received Duplicate message

Node 13 Received Duplicate message

Node 25 Received Duplicate message

NODE:41 Received from node 27 with location (75.000000 ,150.000000 ,0.000000)
Threshold Distance is: 70.933200
Distance from the previous Node: 50.000000
Node 41 Message Discarded(distance < threshold)

Node 12 Received Duplicate message

Node 18 Received Duplicate message

Node 32 Received Duplicate message

Node 40 Received Duplicate message

Node 11 Received Duplicate message

Node 39 Received Duplicate message

Node 6 Received Duplicate message

Node 24 Received Duplicate message

NODE:48 Received from node 27 with location (75.000000 ,150.000000 ,0.000000)
Threshold Distance is: 70.933200
Distance from the previous Node: 75.000000
The Node 48 Sends message

Node 5 Received Duplicate message

Node 17 Received Duplicate message
Node 31 Received Duplicate message

NODE:47 Received from node 27 with location (75.000000 ,150.000000 ,0.000000)
Threshold Distance is: 70.933200
Distance from the previous Node: 79.056942
The Node 47 Sends message

Node 4 Received Duplicate message

Node 10 Received Duplicate message

Node 38 Received Duplicate message

Node 46 Received Duplicate message

Node 23 Received Duplicate message
Node 16 Received Duplicate message

Node 30 Received Duplicate message

.
.
.
.
.
.
.
.

NODE: 0
Data Originated 0 Data Transmitted 0 Data Received 30
NODE: 1
Data Originated 30 Data Transmitted 30 Data Received 0
NODE: 2
Data Originated 0 Data Transmitted 0 Data Received 30
NODE: 3
Data Originated 0 Data Transmitted 0 Data Received 30
NODE: 4
Data Originated 0 Data Transmitted 30 Data Received 30
NODE: 5
Data Originated 0 Data Transmitted 30 Data Received 30

.
.

.
.
.
NODE: 43
Data Originated 0 Data Transmitted 19 Data Received 29
NODE: 44
Data Originated 0 Data Transmitted 21 Data Received 29
NODE: 45
Data Originated 0 Data Transmitted 20 Data Received 29
NODE: 46
Data Originated 0 Data Transmitted 22 Data Received 29
NODE: 47
Data Originated 0 Data Transmitted 21 Data Received 29
NODE: 48
Data Originated 0 Data Transmitted 22 Data Received 29 Execution time : 0.6570 sec

5.3 Result of EBSF GUI

The following figure shows the simulator GUI where we can see the node 35 communicating with other nodes for message transmission.

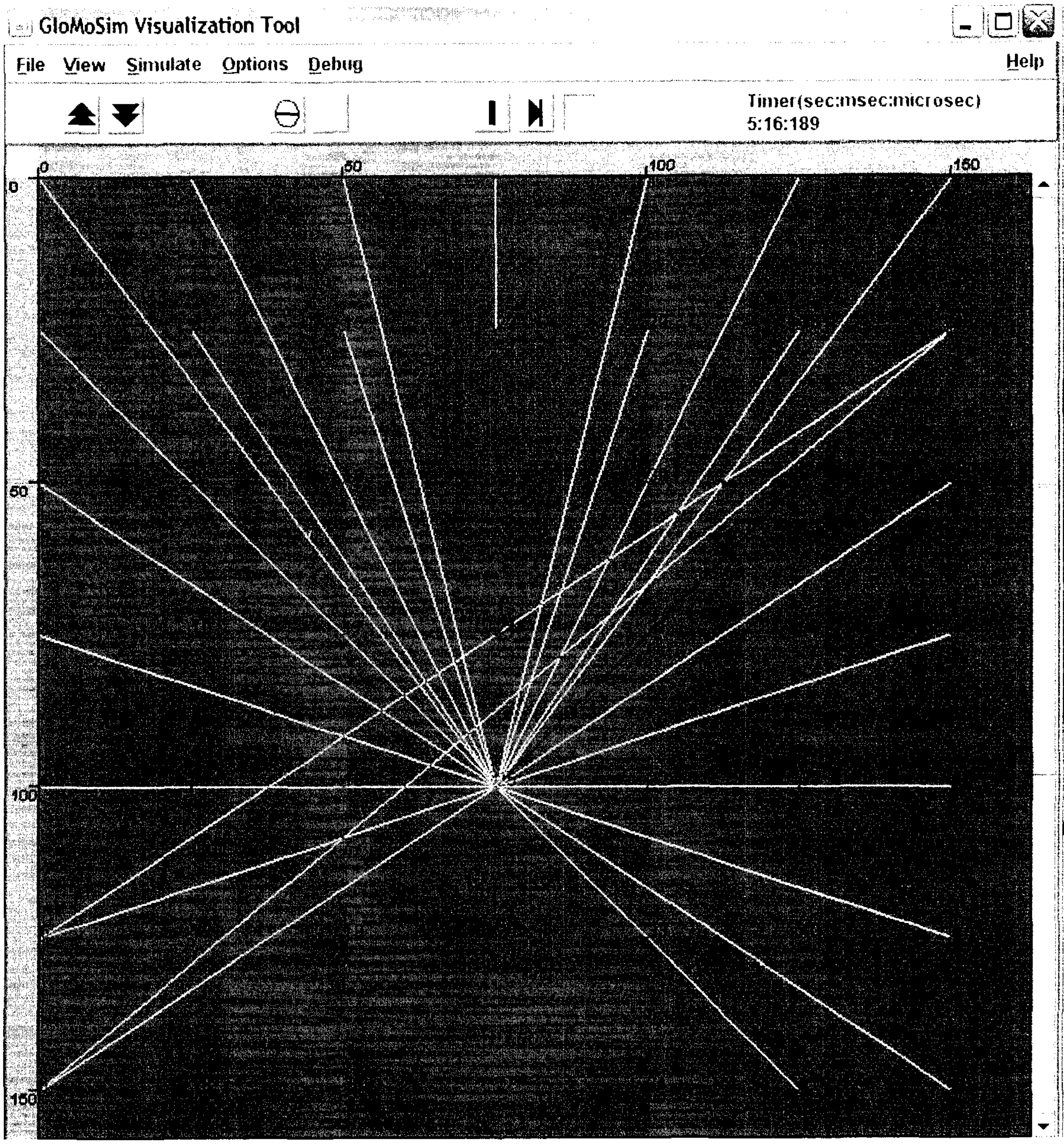


Figure 11 GUI of EBSF

5.4 Performance Analysis

We consider number of nodes = 49

Total number of transmissions for flooding = 1470

Total number of transmissions for ESBF with threshold =0.45*R = 1117

Total number of transmissions for ESBF with threshold =0.60*R = 822

Total number of transmissions for ESBF with threshold =0.75*R = 540

Total number of transmissions for enhanced EBSF = 405

Where R is the transmission range of each node.

Performance statistics:

Table 1. Performance statistics

NODE	FLOODIN G	ESBF threshold =0.45*R	ESBF threshold =0.6*R	ESBF threshold =0.75*R
0	30	0	0	0
1	30	30	30	30
2	30	0	0	0
3	30	0	0	0
4	30	30	30	0
5	30	30	30	30
6	30	21	7	8
7	30	0	0	0
8	30	0	0	0
9	30	0	0	0
10	30	30	0	0
11	30	30	30	0
12	30	30	30	30
13	30	20	8	10
14	30	30	0	0
15	30	0	0	0
16	30	30	0	0
17	30	30	0	0
18	30	30	30	30
19	30	30	30	30
20	30	22	9	8
21	30	30	30	0
22	30	30	30	0
23	30	30	30	0
24	30	30	30	30
25	30	30	30	30
26	30	18	14	4
27	30	22	16	2

28	30	30	30	30
29	30	30	30	30
30	30	30	30	30
31	30	30	30	30
32	30	22	16	10
33	30	22	16	9
34	30	25	18	8
35	30	20	13	12
36	30	22	10	9
37	30	26	9	6
38	30	27	12	6
39	30	21	16	8
40	30	24	17	7
41	30	27	19	12
42	30	25	17	13
43	30	25	19	13
44	30	28	21	10
45	30	26	20	9
46	30	22	22	12
47	30	25	21	12
48	30	27	22	22

5.5 Performance Analysis using Bar Chart

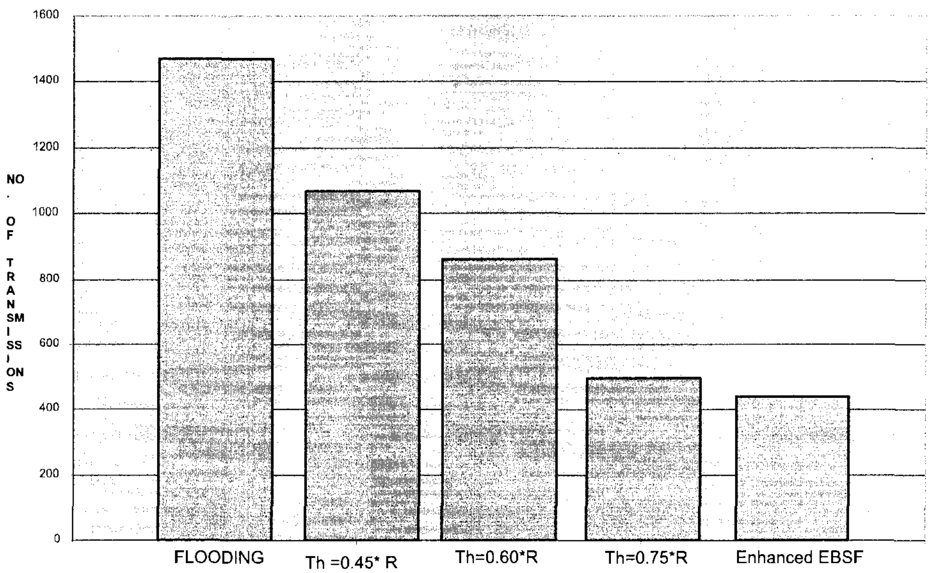


Figure 12 Bar-chart analysis for different threshold values

* The efficiency of EBSF remains very high even in large networks.

* Our Simulation suggests that there is efficient reduction in data transmission.

* Higher the threshold value, more optimized results.

* The threshold value can be tuned to show the performance enhancements.

CHAPTER 6

CONCLUSION AND FUTURE WORK

Building efficient protocols for ad hoc networks is challenging due to dynamic nature of the nodes. Efficient Broadcasting by Selective Forwarding has a number of advantages over other approaches considered in the literature. The threshold value can be tuned to enhance the performance. The Higher the threshold value, the more optimized the results. It does not impose any bandwidth overhead and also reduces the power consumption drastically. The efficiency of EBSF remains very high even in large networks. Overall, the proposed protocol shows that broadcasting can be enhanced greatly by choosing only an optimal set of nodes for transmission and thus avoiding redundant transmissions, and at the same time ensuring data delivery to all the nodes in the network.

This protocol could be integrated with any routing protocol for finding a route in mobile ad-hoc networks with minimal power consumption, and without imposing any bandwidth overhead.

BIBLIOGRAPHY

[1] Y.B. Ko and N.H. Vaidya, Location-aided routing in mobile ad hoc networks, Technical report 98-012, Texas A&M University (1998).

[2] Xiaohu Chen, Michalis Faloutsos, Srikanth Krishnamurthy, Distance Adaptive (DAD) broadcasting for ad hoc networks, MILCOM 2002 Proceedings, Oct. 2002 Volume 2.

[3] Vamsi K Paruchuri, Arjan Durresi, Raj Jain, Optimized Flooding Protocol for Ad hoc Networks, IEEE Transactions on Parallel and Distributed Systems,vol. 15, No. 11, Nov.2004,1027-1040.

[4] S. Y. Ni et al. The Broadcast Storm Problem in a Mobile Ad Hoc Network. ACM MOBICOM, pp. 151-162, Aug'1999.

[5] N. Alon, A. Bar-Noy, N. Linial, and D. Peleg. A lower bound for radio broadcast. J. Comput. Syst. Sci., vol. 43, pp. 290.298, Oct. 1991.

[6] I. Gaber and Y. Mansour. Broadcast in radio networks. In Proc. 6th Annual. ACM-SIAM Symp. Discrete Algorithms, San Francisco, CA, Jan. 1995, pp. 577.585.

[7] S. Guha and S. Khuller. Approximation algorithms for connected dominating sets. In Proceedings of European Symposium on Algorithms (ESA), 1996.

[8] H. Lim and C. Kim. Multicast tree construction and flooding in wireless ad hoc networks. In Proceedings of the ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM), 2000.

[9] Jie Wu and Fei Dai. Broadcasting in Ad Hoc Networks Based on Self-Pruning. In Proceedings of IEEE INFOCOM 2003, San Francisco, CA.

[10] W. Peng and X. Lu. On the reduction of broadcast redundancy in mobile ad hoc networks. In Proceedings of MOBIHOC, 2000.

[11] W. Peng and X. Lu. AHBP: An efficient broadcast protocol for mobile ad hoc networks. Journal of Science and Technology - Beijing, China, 2002.

[12] A. Qayyum, L. Viennot, and A. Laouiti. Multipoint relaying: An efficient technique for flooding in mobile wireless networks. Technical Report 3898, INRIA - Rapport de recherche, 2000.

[13] J. Sucec and I. Marsic. An efficient distributed networkwide broadcast algorithm for mobile ad hoc networks. CAIP Technical Report 248-Rutgers University,September 2000

[14]. Xiaohu Chen, Michalis Faloutsos, Srikanth V. Krishnamurthy Power Adaptive Broadcasting with Local Information in Ad hoc networks Proceedings of the 11th IEEE International Conference on Network Protocols (ICNP'03)

[15]. Xiaohu Chen, Michalis Faloutsos, Srikanth Krishnamurthy Distance ADaptive (DAD) Broadcasting for Ad Hoc Networks Department of Computer Science, University of California, Riverside IEEE Transactions on Parallel and Distributed Systems, Vol. 15,No. 10, Oct. 2004, 908-920.

[16] E. Royer and C.K.Toh, " A Review of current Routing Protocols Protecls for Ad Hoc Networks", IEEE Personal communication magazine, April 2002.

[17] Glomosim, Glomosim References. Installation of Glomosim 2.03 in Windows xp. Introduction to Mobile Networks, Simulations using Glomosim. Miscellaneous References WWW.cs.ndsu.edu/~ahmed/Glomosim.html

[18] http://pcl.cs.ucla.edu/projects/Glomosim/

[19] www.sm.luth.se/csee/courses/smd/161_wireless/glomoman.pdf

[20] B. Williams and T. Camp. Comparison of broadcasting techniques for mobile ad hoc networks. In Proceedings of the ACM Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC), pages 194–205, 2002.

VITA

Graduate College
University of Nevada, Las Vegas

Balaji ashok Sathyanarayanan

Address:
4235, Cottage Circle, Apt # 4
Las Vegas, NV 89119, USA.

Degree:
Bachelor of Engineering, Computer Science, 2004
University of Madras, India

Thesis Title: Efficient Broadcasting by Selective Forwarding

Thesis Examination Committee:
Chairperson, Dr. Ajoy Datta, Ph. D.
Committee Member, Dr. John T Minor, Ph. D.
Committee Member, Dr. Yoohwan Kim, Ph. D.
Graduate College Representative, Dr. Venkatesan Muthukumar, Ph. D.