Fall 2009

Localization of nodes in wired and wireless networks

Hindu Kothapalli

LOCALIZATION OF NODES IN WIRED

AND WIRELESS NETWORKS

by

HINDU KOTHAPALLI

A THESIS

Presented to the Faculty of the Graduate School of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

2009

Approved by

Jagannathan Sarangapani, Advisor
Maciej Zawodniok, Co-advisor
Yahong Rosa Zheng
Sahra Sedighservestani

PUBLICATION THESIS OPTION

This thesis consists of the following articles that have been submitted for publication as follows:

Pages 3-33 are intended for submission to the *Journal of Applications of Artificial Intelligence*

Pages 34-84 are intended for submission to the *International Journal of Distributed Sensor Networks*

ABSTRACT

This thesis focuses on the implementation of algorithms for localization of nodes in wired and wireless networks. The thesis is organized into two papers. The first paper presents the localization algorithms based on time of arrival (TOA) and time difference of arrival (TDOA) techniques for computer networks such as the Internet by using round-trip-time (RTT) measurements obtained from known positions of the gateway nodes. The RTT values provide an approximate measure of distance between the gateway nodes and an unknown node. The least squares technique is then used to obtain an estimated position of the unknown node.

The second paper presents localization of an unknown node during route setup messages in wireless ad hoc and sensor networks using a new routing protocol. A proactive multi-interface multichannel routing (MMCR) protocol, recently developed at Missouri S&T, was implemented on the Missouri S&T motes. This protocol calculates link costs based on a composite metric defined using the available end-to-end delay, energy utilization, and bandwidth, and it chooses the path that minimizes the link cost factor to effectively route the information to the required destination. Experimental results indicate enhanced performance in terms of quality of service, and implementation of this protocol requires no modification to the current IEEE 802.11 MAC protocol. Received signal strength indicator (RSSI) values are recorded from the relay nodes (gateway nodes) to the unknown node during route setup messages. The location of the unknown node is estimated using these values with some a priori profiling and the known positions of the relay nodes as inputs to the least squares technique.

ACKNOWLEDGMENTS

I would like to express my gratitude to my advisor, Dr. Jagannathan Sarangapani, for his advice, guidance, and encouragement throughout my graduate studies and particularly during this research. Also, I would like to express my deepest thanks to my co-advisor, Dr. Maciej Zawodniok, for his invaluable contributions and suggestions that helped me complete this work. The financial assistance provided in the form of Graduate Research Assistantship from Airforce Research Laboratory grant is thankfully acknowledged.

I also would like to thank Dr. Sahra Sedighservestani and Dr. Yahong Rosa Zheng for their time and effort in serving as committee members and reviewing this thesis. Special thanks go to my colleagues Jake Hertenstein, Mohammed Rana Basheer, and Priya Kasirajan for their advice and for hours of discussion.

Lastly, and most importantly, I am indebted to my parents for their patience with me during my endeavors. Without their support, I would not have been successful.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

1. INTRODUCTION

Location service is a fundamental building block of many emerging computing/networking paradigms. For example, in pervasive computing, knowing the location of the computers and printers in a building allows a computer to send a printing job to the nearest printer. In sensor networks, the nodes must know their locations in order to detect and record events, and to route packets using geometric routing.

Manual configuration is one way to determine the location of a node. However, this is unlikely to be feasible for large-scale deployments where nodes move often. Another possibility is GPS; however, this option is costly in terms of both hardware and power requirements. Since GPS requires line of sight between the receiver and satellites it consumes additional energy and is too expensive to integrate on hundreds of energy-constrained sensor nodes. Furthermore, it may not work well in buildings or in the presence of obstructions such as dense vegetation or mountains that block direct view of the GPS satellites. The use of sensor nodes to localize the indoor environment provides a less expensive alternative.

Localization is important both for wired and wireless networks. If the position of the nodes is known, the network can be more effectively analyzed, and its performance can be optimally enhanced. For instance, the Internet continues to grow, and it could be much more efficient if localization techniques were applied to estimate the position of the nodes. Increasingly, Internet hosts must be able quickly and efficiently to determine their distance from one another in terms of metrics such as latency or bandwidth.

Recent advances in wireless communications and electronics have permitted the development of low-cost, low-power, and multifunctional sensors with small form factor to communicate over short distances. Cheap, smart sensors, networked through wireless links and deployed in large numbers provide opportunities for monitoring and controlling homes, cities, and environment.

Knowing the position of the sensor nodes is important for several reasons: First, self-configuration and self-organization are key entities for robustness, and they can be easily supported if position information is available. In addition, fully covered sensor networks permit energy aware geographic routing. Further, information received with no indication of source location is generally useless. Finally, in many applications the position itself is the information of interest. For example, applications such as tracking endangered species and tracking wild fires demand an exact, or at the least an approximate, position of the sensors. In security applications such as surveillance sensor networks, knowledge of the location enhances security. Mobile sensor nodes are only controlled if their locations are known. Home automation and energy conservation depend on location-based routing. Locations are also helpful in inventory management and habitat monitoring. Thus, localization of nodes in wireless networks is becoming a necessity in many applications where source of incoming measurements must be located as precisely as possible, even when the nodes are mobile.

Many localization algorithms require distance information to estimate the position of unknown devices. This information can be obtained by measuring the received signal strength indicator (RSSI), the time of arrival (ToA), or the time difference of arrival (TDoA).

In addition to providing mere connectivity information, communication between two nodes often permits the extraction of information about the geometric relationship between nodes. Using elementary geometry, this information can be used to derive information about node positions. When distances between entities are used, the approach is called *lateration*. For lateration in a plane, the simplest option is for a node to have precise distance measurements to three gateway nodes. Extension to a three-dimensional space is trivial. Using distances and gateway node positions, the unknown node’s position must be at the intersection of three circles around the gateway. In reality, however, distance measurements are never perfect, and the intersection of these circles will not generally be a single point.

To overcome these imperfections, distance measurements from more than three reference nodes can be used, resulting in a multilateration problem. To use multilateration, distances to gateway nodes must be estimated. The characteristics of any communication, whether wired or wireless, are partially determined by the distance between sender and receiver, and if these characteristics can be measured at the receiver they can serve as an estimator of distance.

Outline of the thesis: This thesis demonstrates how the unknown node in wired and wireless networks can be localized. In the case of wired networks, when the RTT values and the position of the gateway nodes are known, the location of an unknown node is estimated using the least squares technique. By contrast, the location of an unknown wireless node is determined by applying RSSI values returned from known relay nodes to the least squares technique.

PAPER

I. LOCALIZATION OF UNKNOWN NODES IN COMPUTER NETWORKS

Hindu Kothapalli and S. Jagannathan
Department of Electrical and Computer Engineering
Missouri University of Science and Technology, Rolla, MO 65401.
hk5y3@mst.edu and sarangap@mst.edu

***ABSTRACT* —**Localization of hardware assets in computer networks is becoming an important goal of many providers. By locating malfunctioning nodes on the Internet, better service can be provided. However, localization of nodes in a computer network is quite difficult due to fewer gateway nodes used for localization for geographically spaced complex Internet and the presence of network traffic which varies significantly with the day and time corrupting the measurements.

In this paper, time of arrival (TOA) and time difference of arrival (TDOA)-based techniques which utilize round-trip time (RTT) measurements from gateway nodes are employed for localization. Based on the RTT values obtained, an approximate measure of distance between the unknown and the gateway nodes is obtained. Various regression techniques are used in order to find a correlation between the RTT measured and the distance obtained. By applying least square algorithm, an accurate estimation of the unknown node is calculated from the gateway node locations. Simulation results verify the performance of the proposed scheme under different topologies, varying number of gateway nodes, and traffic.

I. INTRODUCTION

Despite the Internet's critical importance in society, very little quantitative information is actually available regarding the structure of the network and its growth. Developing a better understanding of the Internet could allow network engineers to further optimize the working of the Internet [1]. Increasingly, Internet hosts must be able to determine their distance quickly and efficiently from one another in terms of metrics such as latency or bandwidth.

For example, to select the nearest multiple web servers in a large-scale network service, distance information would be useful. In overlay network multicast applications, such as peer-to-peer file sharing sites like the Napster and Gnutella, distributed content services store multiple copies of the same item on the Internet. The user can be directed to any one of several online copies, but it is beneficial to direct the user to the nearest system with the service available. Even when all other things are not equal, such as the case where different web servers have different response times, it is still useful to include distance to each candidate host as one of several criteria for making a selection [2].

Moreover, location awareness, including the ability to locate a wireless network user, is an important requirement for many applications. For example, localization can be used in the E911 service to trace the location of the caller quickly [20] in the case of emergency. Other applications that demand this function include cyber crime detection, lost mobile phone tracing, and maintenance of switches. With so many important networking applications that require location, estimation of the location of an unknown node could prove vital in the future.

Localization is important for facilitating location-based services. It determines the location of one or more devices based on certain measurements. Location could be expressed as the coordinates of the source or target, which may be in two or three dimensions. These coordinates could include information such as latitude and longitude where the source is located. These network coordinates provide a practical and efficient way to estimate network distances among computers in the network.

Location information is complex, not merely a set of Cartesian coordinates [8]. A location service collects; stores, and provides access to location information. Other applications may then obtain this information from the location service. The location service may even be integrated with other systems, providing related data to form a geographic information system (GIS).

Location service is a fundamental building block of many emerging computing and networking paradigms. In pervasive computing [9, 10] knowing the locations of the computers and the printers in a building allows a computer to send a printing job to the nearest printer. In sensor networks, the sensor nodes must know their locations to detect and record events and to route packets using geometric routing [11]. Manual configuration is one method to determine the location of a node. However, this method is unlikely to be feasible for large-scale deployments and scenarios.

Location information is useful in both fixed and mobile networks where this information is often used for management purposes. Source localization [31] determines a source location through a number of receivers that capture the signal radiated from a source. Passive source localization [32] has been applied in many areas such as wireless communications, geo-location, radar, underwater sonar, and sensor. Localization is important when the exact location of certain fixed or mobile devices is uncertain. One example is in the supervision of humidity and temperature in forests and fields, where thousands of sensors are deployed by an aircarft, giving the operator little or no control over the precise location of each node. An effective localization algorithm can use all available information from the wireless sensor nodes to infer the position of individual devices.

Emerging peer-to-peer overlay network applications can also use distance information to make the overlay network distance-aware, distance information is given high priority when constructing such networks [3]. One of the main challenges in design of network infrastructure is to balance the tradeoffs between providing individual services to each client and at the same time making efficient use of networked resources. This balance enables the infrastructure to accommodate more services and clients and to respond better to a growing number of clients. Quick and efficient location of desired resources at specific network locations will permit effective use of network resources; thus the techniques that permit such localization have become important [4].

Similarly, service providers must know the location of network nodes in order to service them better. Unfortunately, due to intense network traffic and a lack of gateway nodes, the localization of nodes is very difficult. In addition, predicting the network distance will greatly reduce the need to measure network performance characteristics such as latency and bandwidth, a process that is time-consuming and impractical due to high overhead.

Several approaches have been developed to retrieve network distance information. Substantial overhead in terms of both delay and network traffic has boosted the popularity of various projects aimed at collecting network distance information and distributing it to various applications [5, 6] based on round-trip-time (RTT) measurements.

One approach to obtain distance information is for the initiating host to measure distance using ping or traceroute [7] functions. Unfortunately, the routes taken by a packet on the Internet vary significantly, causing an increase in the RTT. The present work proposes a means to provide distance information in terms of latency (e.g., round-trip delay). Latency is the easiest information to provide, and it is both useful and easy to measure. Therefore, it is not surprising that RTT plays a significant role in several protocols and applications, such as overlay network construction protocol, peer-to-peer services, and proximity-based sewer redirection.

The localization methods can be classified as range-based methods if an estimated distance between two nodes is computed. Range-based methods exploit information about the distance to neighboring nodes. Although the distances cannot be measured directly, they can, at least theoretically, be derived from measuring the time-of-flight between nodes for a packet or from signal attenuation. The simplest range-based method is to require knowledge of distances to three nodes with known positions, called reference nodes, and then use triangulation. However, more advanced methods are available requiring milder assumptions.

Many techniques have been developed in the past 30 years to locate a source using a set of measurements from the gateway receivers. Three of the most popular techniques are angle of arrival (AOA), time of arrival (TOA) and time difference of arrival (TDOA) based schemes. The AOA or direction of arrival (DOA) scheme determines the target location by triangulation. It locates the target by the intersection of two lines of bearing from two receiver positions which are at a known distance apart. This technique requires a minimum of two receivers to determine a position in two dimensions. For three-dimensions, measurements for length, azimuth, and two angles are needed to specify a precise position [12]. Usually, the AOA estimate is improved by exploiting the redundant information of multiple AOA estimates. This method has been widely used in radar tracking, surveying and vehicle navigation systems. The major drawback of AOA scheme is its sensitivity to signal blockage and multipath reflection, especially in urban areas. Most AOA algorithms are highly complex because of the need to measure, store, and use array calibration data and their computationally intensive nature.

By contrast, the TOA scheme determines the location based on speed and propagation time where speed is considered to be a constant. Therefore, the distance from the source to the receiver is directly proportional to propagation time. Thus, the intersection of the radial distance from three receivers would then constitute its location.

Since the propagation speed of the signals is quite high (equals to two third of the speed of light in the copper wire as the propagation medium), time measurements must be very accurate to avoid large errors. For example, a localization accuracy of 1m requires timing accuracy on the level of 5 nanoseconds. Hence, this method requires synchronous clocks between the receivers and the source. Also, the source must be labeled with a time stamp in order for the receivers to calculate the distances traveled by the signals.

The TDOA scheme determines the relative position of the source by examining the time differences in the arrival of signals from the source at multiple receivers, rather than using absolute arrival times. This method is often referred to as a hyperbolic system because the time difference of two receivers is converted to a constant distance difference of two receivers and defines a hyperbolic curve. In two-dimensions, the intersection of two hyperbolic curves renders the source position. The accuracy of this method is a function of the relative distance between the source and receiver locations.

These signal-propagation-based time-measuring positioning systems provide an acceptable location estimation since the signal propagation time is less affected by environmental factors than are with other methods such as AOA and received signal strength (RSS) [13,14]-based schemes. The TOA and TDOA are well-known propagation time-based methods. The global positioning system (GPS) is the most widely used system, based on the TDOA concept [15]. However, these methods require highly resolute timing information and precise time synchronization to reach an accuracy level of 1cm [16, 14].

The following section presents, a brief overview of the background work done in localization.

II. BACKGROUND

Practical distance estimates play an essential role in the selection of nearest server. Many widely used Internet services are replicated (or mirrored) in multiple physical locations. The goal of this replication is to provide users with faster access to content by allowing them to select nearby copies and avoid congested paths or servers. The framework presented here defines as nearest to a client the server with the lowest RTT from that client.

Several attempts [3-6, 17, 18, 24, 25, 27, 33-38] have been made to determine the location of Internet network topologies with attempting to actually to construct its topology in order to analyze it. Most existing methods of location determination require [1, 5, 12-18, 21-25, 31-36] reference nodes with known positions around an unknown node. Here, several network parameters and methods have been used to infer the network topology using the range-based methods described above. Although RSSI and AOA can be used only for wireless networks, TOA and TDOA can be used both for wired and wireless networks.

Some other measurement-based approaches [17] have attempted to determine network topology based on several network properties, such as traffic, RTT, and packet dropping rate. Other methods [18] rely on the locations of remote nodes relative to a center node using relative maps of the nodes on one or more communication paths between them. One of the fundamental weaknesses of these schemes is their limited accuracy. Another approach for determining the location [19] is to determine the delay between the source and the destination and then find a correlation between geographical distance and delay. The basic assumption of these schemes is that hosts with similar network delays to some fixed probe machines tend to be located near each other.

Localization with TOA and TDOA usually involves of two steps: First, the TOA and TDOA are obtained from RTT measurements. Second, these parameters are used to obtain the final location estimate using some sort of nonlinear optimization problem [21]. Many algorithms have been developed to improve the accuracy of location [13-15, 18, 21-23, 25-27]. On approach uses maximum likelihood, but this method is computationally costly to implement with grid searches.

This paper provides an approach to the problem of network localization in wired networks according to which certain nodes, called gateway nodes know their locations, and are used to determine the location of other nodes by measuring the distances to their neighbors. It presents a location technique based on distance measurements provided by the TOA and TDOA, which are in turn, based on the RTT measurements between the unknown node and the gateway or reference nodes. Although the TOA and the TDOA methods have been shown to be effective, they do not effectively maintain time synchronization. The RTT-based method requires no time synchronization even though the delay arising from the node makes the exact measurement difficult.

An important characteristic of this system presented here is its simplicity. The system is divided into two subsystems: the ranging and the positioning subsystems. The former estimates the distance between a unknown node and the gateway nodes, and the latter calculates the unknown node's position based on these estimations using least square approximation [22]. The gateway nodes and landmarks are used to determine the position of the unknown node relative to them. This work also addressed the number of gateway nodes necessary to determine accurately the location of an unknown node. The main contributions of this work include: (1) measuring RTTs between the nodes in the network, (2) estimating distance information using these RTT values, (3) calculating TOA and TDOA to estimate the location of the unknown node, and (4) minimizing the error in finding the position of the unknown node using the least square method.

III. METHODOLOGY

Node localization using TOA and TDOA measurements involves two steps. First, the TOA and TDOA between receivers are estimated using the RTT-based time delay technique. The estimated time and time difference are then converted to range and range difference measurements respectively among the receivers, resulting in a set of nonlinear hyperbolic equations.

This paper seeks to evaluate the feasibility of determining location using a least square approximation method. With delay measurements, the location of the unknown node can be determined relative to a set of known nodes. The simulations were performed using Network Simulator 2 (NS2) software, [28, 29] a discrete event network simulator. This software is used to simulate routing and multicast protocols, among others, and it is heavily used in network research both in academics and industry. It supports an array of popular network protocols, offering simulation results for wired and wireless networks alike. It can be also used as a limited-functionality network emulator.

A. Ranging System

A packet traverses many links on its way from source to destination, and several parameters of each link (e.g., propagation latency, available bandwidth, queuing delay, and packet loss) contribute to overall end-to-end delay. These parameters are generally unknown and can fluctuate unpredictably over time due to network traffic. Fig. 1 shows the delays incurred when traversing a packet between two nodes. Therefore, consistently useful quantitative measurements of host-to-host performance, particularly those that can predict future performance, are extremely challenging.

Although the primary goal of Internet architecture is not to facilitate performance measurements between end hosts [5], several tools to measure RTT exploit features in the Internet Protocol (IP). One of the common tools to measure path RTT is the ping command. This tool works by sending an Internet Control Message Protocol (ICMP) packet, usually referred to as a *probe*, which forces the end host to reply. The RTT, then, is the elapsed time between the sending of the ICMP packet and the reply. The ping command is often used to determine the host reachability. It also provides a way to observe the dynamics of the RTTs along a path to determine any path anomalies. In addition, ping provides statistics on probes (e.g., minimum, average, and maximum RTTs) as well as probe losses. The RTT and the distance between any two hosts can be calculated using TOA as

$$\text{Round Trip Time} = 2 \times \text{Time of Flight} \quad (1)$$

$$\text{Distance} = \frac{RTT}{2}c = \frac{(t4-t1)-(t3-t2)}{2}c \quad (2)$$

where c denotes the speed of signal transmission, which in a copper wire-based network, the speed of travel is two thirds the speed of light [30], or $c = 2\times10^5 km/\sec$. Hence, in 1 millisecond, the signal travels 200 km. The distance is calculated based on this assumption.

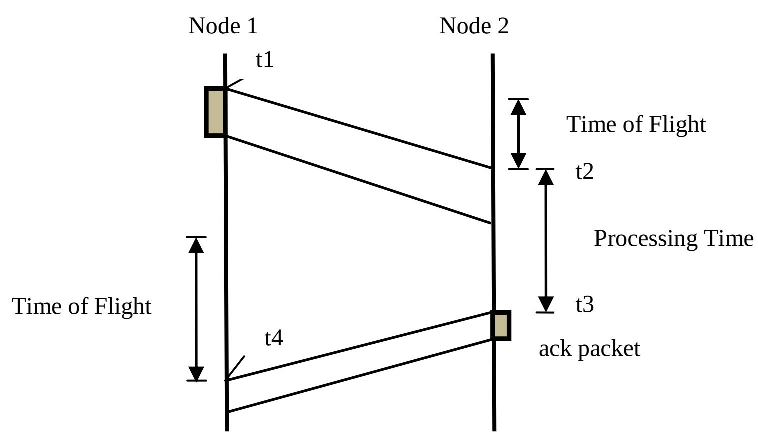


Figure 1. Delays incurred when exchanging packets

By contrast, to determine RTT and distance using TDOA, the range difference is calculated by obtaining the time difference between the signals received at multiple gateway nodes from the unknown node as

$$t_{ij} = (t_i - t_j) \quad (3)$$

where t_{ij} is the time difference between the signals received at gateway nodes i and j from the unknown node. Hence, the range difference can be calculated as

$$d_{ij} = (d_i - d_j) = ct_{ij} \ , \tag{4}$$

where c is the speed of the signal in copper wire, as given in equation (4).

B. Positioning System

Once the distances are obtained, the location of the unknown node can be computed by using least squares estimation. Since a very large geographical network is considered and the distance between the nodes is in the range of hundreds of kilometers, hence the path between the nodes taken is assumed to be a straight line. Let the position of the unknown node be (x, y). Let (x_i, y_i) be the positions of the reference nodes where i =1, 2….N. Let the first reference node be placed at (0, 0). The distance is then given by

$$d_1^2 = (x_1 - x)^2 + (y_1 - y)^2 \tag{5}$$

$$d_2^2 = (x_2 - x)^2 + (y_2 - y)^2 \tag{6}$$

$$d_i^2 = (x_i - x)^2 + (y_i - y)^2 \ . \tag{7}$$

Therefore,

$$d_2^2 - d_1^2 = x_2^2 - 2x_2 x + y_2^2 - 2y_2 y \tag{8}$$

$$d_3^2 - d_1^2 = x_3^2 - 2x_3 x + y_3^2 - 2y_3 y \ , \tag{9}$$

Equations (6) and (7) can be represented in matrix form:

$$\begin{pmatrix} x_2 & y_2 \\ x_3 & y_3 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{2} \begin{pmatrix} k_2^2 - d_2^2 + d_1^2 \\ k_3^2 - d_3^2 + d_1^2 \end{pmatrix} \tag{10}$$

Equation (8) can be rewritten in the form of $Ax = B$ where

$$A = \begin{pmatrix} x_2 & y_2 \\ x_3 & y_3 \\ \vdots & \vdots \\ x_i & y_i \end{pmatrix} \qquad B = \frac{1}{2} \begin{pmatrix} k_2^2 - d_2^2 + d_1^2 \\ k_3^2 - d_3^2 + d_1^2 \\ \vdots \\ k_i^2 - d_i^2 + d_1^2 \end{pmatrix} . \tag{11}$$

with $k_i^2 = x_i^2 + y_i^2$. The Least square solution is obtained as

$$x = (A^T A)^{-1} A^T B . \tag{12}$$

By contrast, the location of the unknown node can then be calculated using the following TDOA scheme as follows. Let the position of the unknown node be (x, y). Let (x_i, y_i) be the positions of the reference nodes where i =1, 2….N. Let the first reference node be placed at (0, 0). The distance is given by

$$d_1^2 = (x_1 - x)^2 + (y_1 - y)^2 \tag{13}$$

$$d_2^2 = (x_2 - x)^2 + (y_2 - y)^2 \tag{14}$$

$$d_i^2 = (x_i - x)^2 + (y_i - y)^2 \ . \tag{15}$$

Thus, the range difference is expressed as:

$$d_{i1} = (d_i - d_1) = ct_{i1} \ . \tag{16}$$

The term d_2^2 can also be written as

$$d_2^2 = (d_{21} + d_1)^2 = (x_2 - x)^2 + (y_2 - y)^2 \tag{17}$$

$$d_{21}^2 + 2d_{21}d_1 = k_2^2 - 2x_2x - 2y_2y \tag{18}$$

$$x_2x + y_2y = \tfrac{1}{2}(k_2^2 - d_{21}^2) - d_{21}d_1 \tag{19}$$

where $k_i^2 = x_i^2 + y_i^2$. Similarly,

$$x_3x + y_3y = \tfrac{1}{2}(k_3^2 - d_{31}^2) - d_{31}d_1 \ . \tag{20}$$

Rewriting these equations in matrix will yields

$$\begin{pmatrix} x_2 & y_2 \\ x_3 & y_3 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} = \tfrac{1}{2}\begin{pmatrix} k_2^2 - d_{21}^2 \\ k_3^2 - d_{31}^2 \end{pmatrix} + d_1\begin{pmatrix} d_{21} \\ d_{31} \end{pmatrix}, \tag{21}$$

which can be generalized as $Ax = B + d_1C$ where the least square solution can be obtained by

$$x = (A^T A)^{-1} A^T (d_1C + B) \ . \tag{22}$$

IV. RESULTS AND DISCUSSIONS

The localization was analyzed using the network simulator, Ns2. The explosive growth of internetworking, and particularly of the Internet, has been accompanied by a wide range of internetworking problems related to routing, resource reservation, and administration. The study of algorithms and policies to address such problems often involves simulation or analysis based on abstraction, or on a model of the actual network structure and applications. The reason is for the use of simulations and models is clear: Networks large enough to be interesting are also expensive and difficult to control; therefore, they are rarely available for experimental purposes. Moreover, assessment of solutions using analysis or simulation is generally more efficient, provided the model is a good abstraction of the real network and application. Therefore, it is remarkable that studies based on randomly-generated or trivial network models are so common, while rigorous analyses of scaled results or their application to actual networks are extremely rare.

This work considered various network topologies using the Georgia Internetwork Topology models, which represent a part of the Internet. To achieve a certain level of confidence in the measured RTT, a large number of probes were transmitted. By considering the average of all received samples, an accurate value of the RTT between the two hosts was obtained. Since the Internet is unpredictable and subject to sudden changes, a single RTT value is not sufficient.

Initially, a network topology with 51 nodes was considered, as shown in Fig. 2 below. This network includes reference nodes, a few random nodes, and one node whose location was unknown. Various levels of network traffic were injected. Transmission control protocol (TCP) sources with constant bit rate (CBR) traffic were introduced with network traffic varying from 10 kbps to 1 Mbps. The median RTT value was calculated from each of these reference nodes, and the values were tabulated.

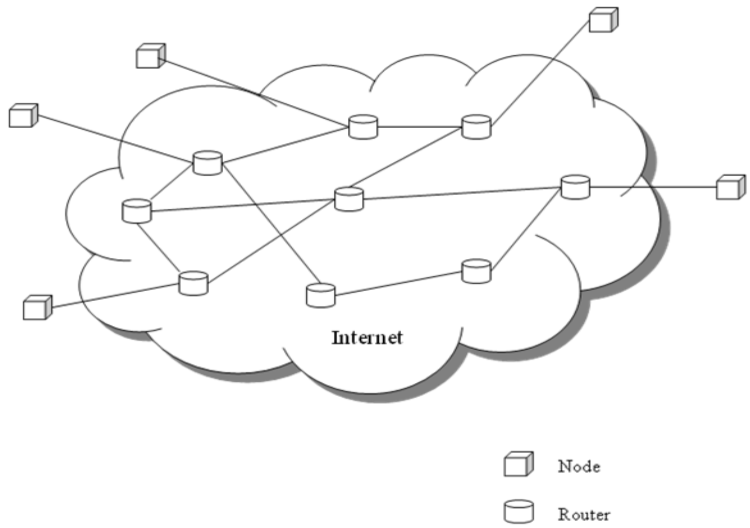
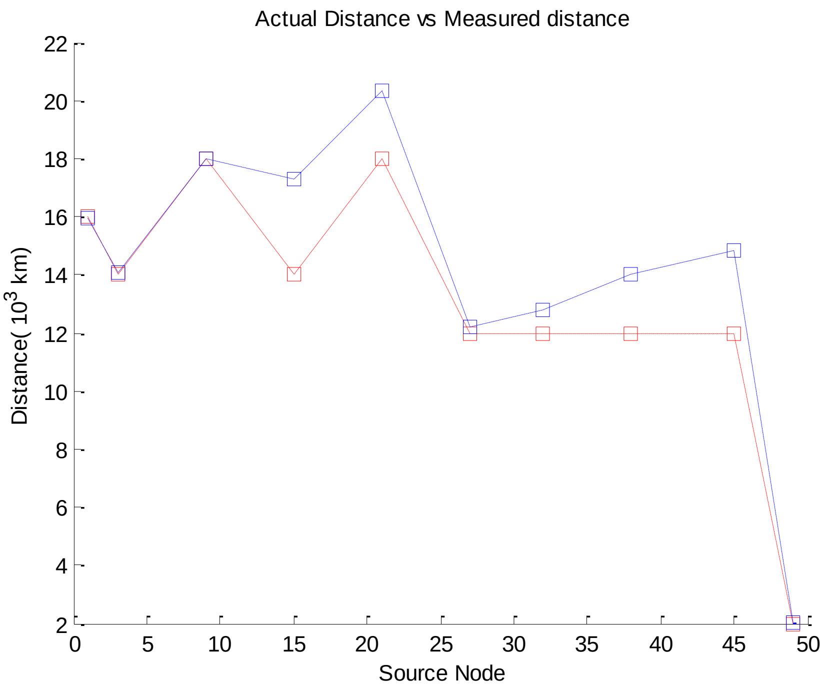


Figure 2. Topology used for determining RTT due to varying traffic

TCP sources and sinks with CBR traffic sources were included, but with no background traffic. Table 1 shows the average values of the RTT with no load.

Table 1. Average RTT and distance measurements for no load

Source Node	Destination Node	Average RTT [milliseconds]	Expected distance [10^3km]	Observed Distance [10^3km]	**Error [%]**
1	51	165.7	16	15.97	**0.1875**
3	51	145.5	14	14.05	**0.3571**
9	51	186.8	18	17.98	**0.111**
15	51	157.7	14	15.27	**9.071**
21	51	190.3	18	18.33	**1.833**
27	51	126.1	12	12.21	**1.75**
32	51	132.0	12	12.80	**6.667**
38	51	133.9	12	12.99	**8.25**
45	51	132.3	12	12.83	**6.916**
49	51	20.02	2	2.002	**0.1**



(Red → Actual Distance Blue → Measured Distance)

Figure 3. Actual distance versus measured distance for no load

As shown in Fig. 3, with no traffic load, there was some variation in the RTT. This variation produced errors in distance estimation even though average values of RTT were acquired by pinging the nodes 20 times. The variation was due to the dynamic routing option typically used by the Internet, wherein the probe packets travel along different paths. In addition, the path taken from source to destination was not necessarily the same as that from destination to source. Hence, the RTT varied depending on traffic and on processing delays at the intermediate nodes in the network. Tables 2, 3, and 4 below show the traffic at various intensities, and the error in the distance calculation.

Table 2. Average RTT and distance measurements for 25% load

Source Node	Destination Node	Average RTT [milliseconds]	Expected distance [10^3km]	Observed Distance [10^3km]	**Error [%]**
1	51	165.7	16	15.97	**0.1875**
3	51	153.0	14	14.80	**5.714**
9	51	195.6	18	18.86	**4.778**
15	51	163.7	14	15.87	**13.357**
21	51	207.9	18	19.09	**6.055**
27	51	126.6	12	12.26	**2.166**
32	51	134.9	12	13.09	**9.083**
38	51	143.9	12	13.99	**16.583**
45	51	132.2	12	12.82	**6.833**
49	51	20.02	2	2.002	**0.1**

Table 3. Average RTT and distance measurements for 50% load

Source Node	Destination Node	Average RTT [milliseconds]	Expected distance [10^3km]	Observed Distance [10^3km]	**Error [%]**
1	51	166.2	16	16.02	**0.1875**
3	51	155.0	14	15.00	**7.1428**
9	51	209.0	18	20.20	**12.22**
15	51	242.2	14	23.72	**69.428**
21	51	268.3	18	26.13	**45.167**
27	51	180.4	12	17.64	**47.0**
32	51	141.9	12	13.79	**14.91**
38	51	145.4	12	14.14	**17.833**
45	51	143.1	12	13.91	**15.91**
49	51	20.02	2	2.002	**0.1**

Table 4. Average RTT and distance measurements for 75% load

Source Node	Destination Node	Average RTT [milliseconds]	Expected distance [10^3km]	Observed Distance [10^3km]	**Error [%]**
1	51	212.3	16	20.03	**25.1875**
3	51	242.3	14	23.73	**69.75**
9	51	309.3	18	30.23	**67.994**
15	51	237.0	14	23.20	**65.7**
21	51	359.0	18	35.20	**95.5**
27	51	167.2	12	16.32	**36.0**
32	51	161.2	12	15.72	**31.0**
38	51	195.5	12	19.15	**59.58**
45	51	168.8	12	16.48	**37.33**
49	51	20.02	2	2.002	**0.1**

These tables show that as the traffic in the network increased, the RTT also increased, resulting in error in distance estimation. The regression curves for various loads were plotted, and they are shown in Figs. 4 through 7.

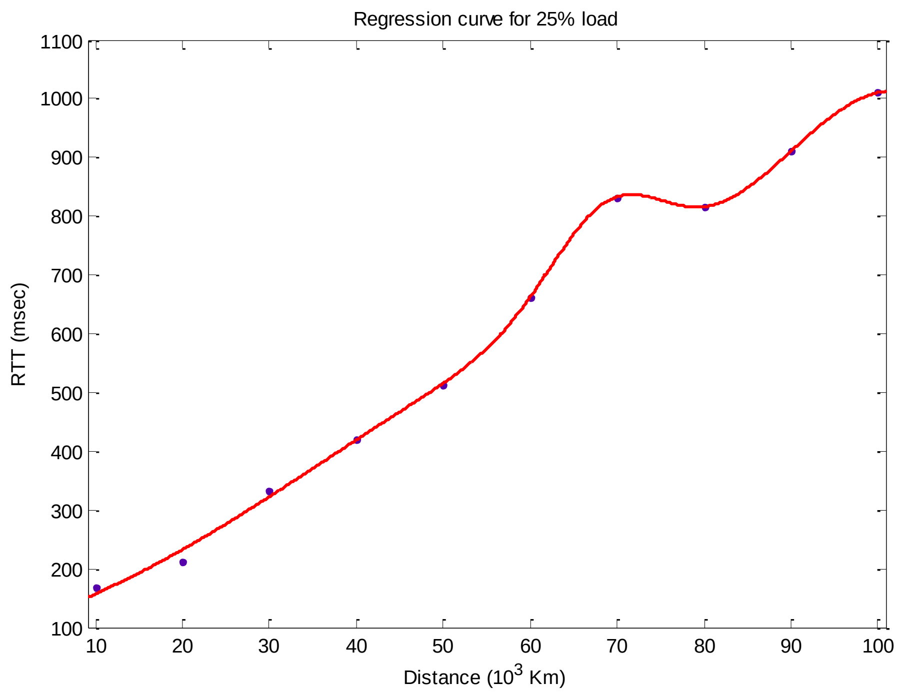


Figure 4. Regression curve for 25% load with gauss fit

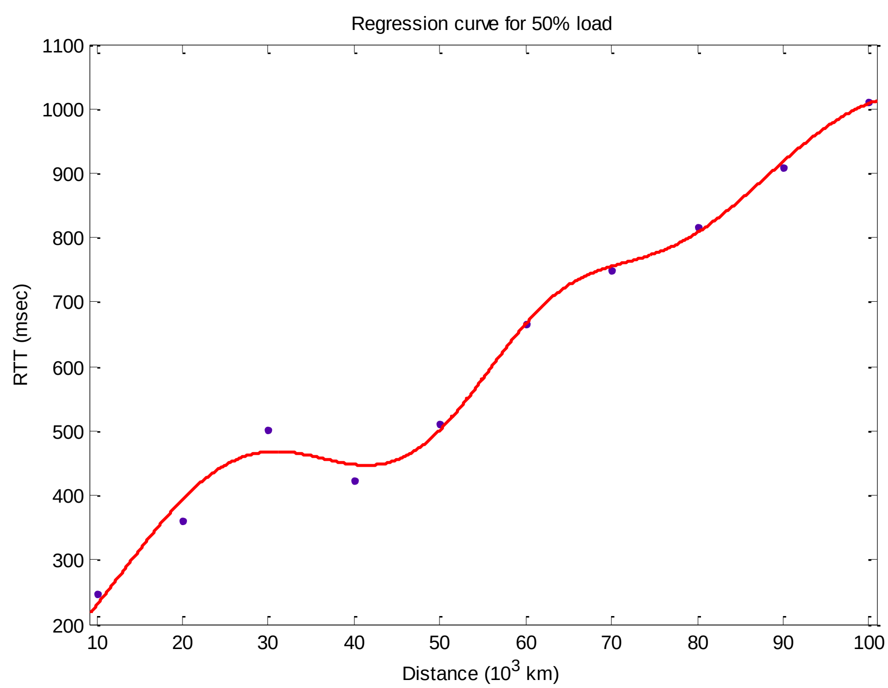


Figure 5. Regression curve for 50% load with gauss fit

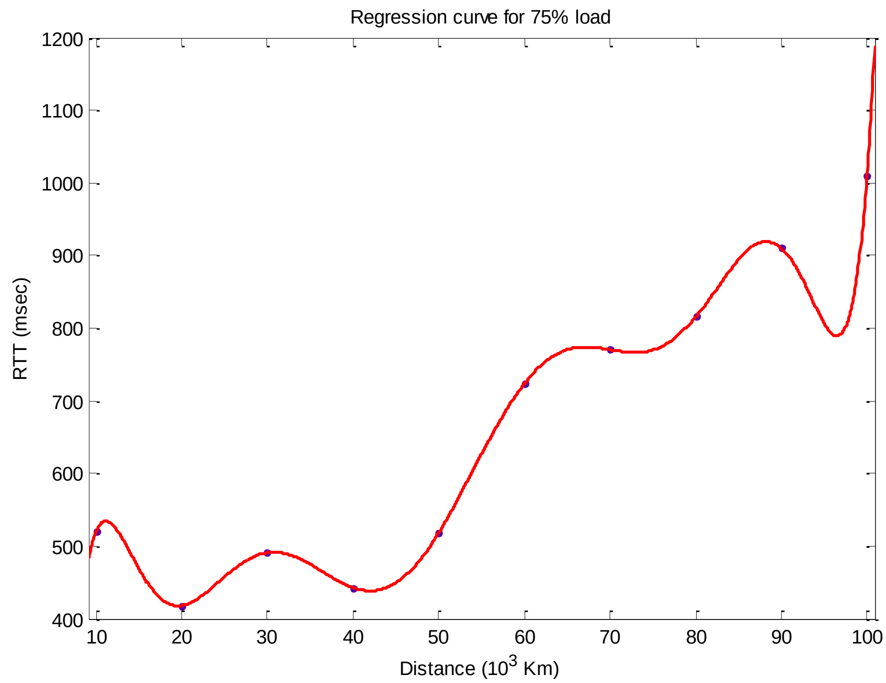


Figure 6. Regression curve for 75% load with gauss fit

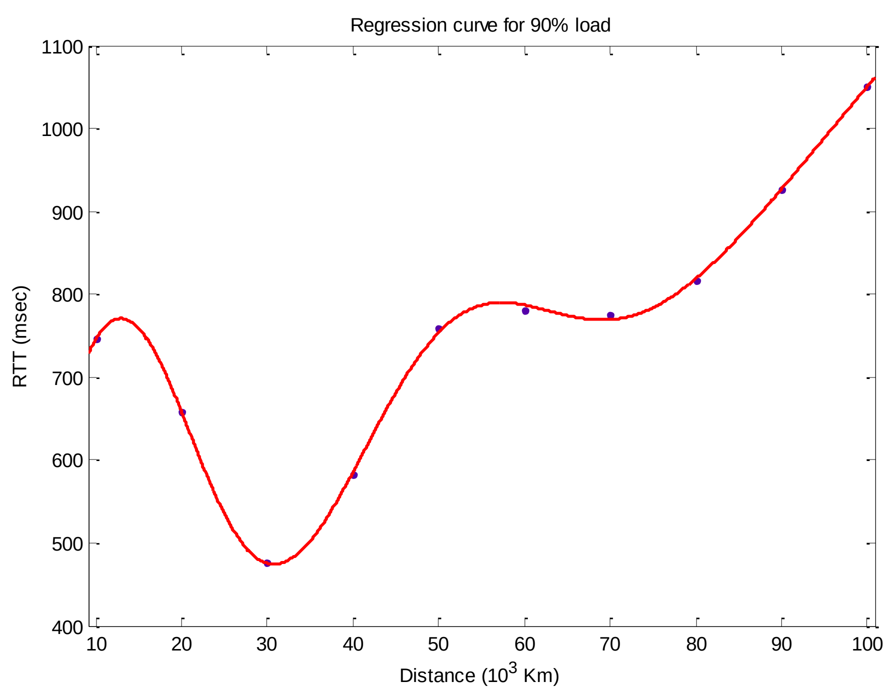


Figure 7. Regression curve for 90% load with gauss fit

These plots demonstrate that as the load on the links increases, the estimation error for distance increases, and thus the graphs shift from linear to nonlinear. To minimize this error, the least square estimation technique is used.

Various topologies were constructed using the Georgia Tech internetwork topology graphs to model network localization, and real time data traffic was applied with the MPEG4 generator to test the conditions for localizability. The topologies generated were the transit stub topology, the hierarchical topology, and the random topology. Each topology consisted of 50 nodes. The various topologies produced, using the network animator, are shown in Figures 8, 9, and 10.

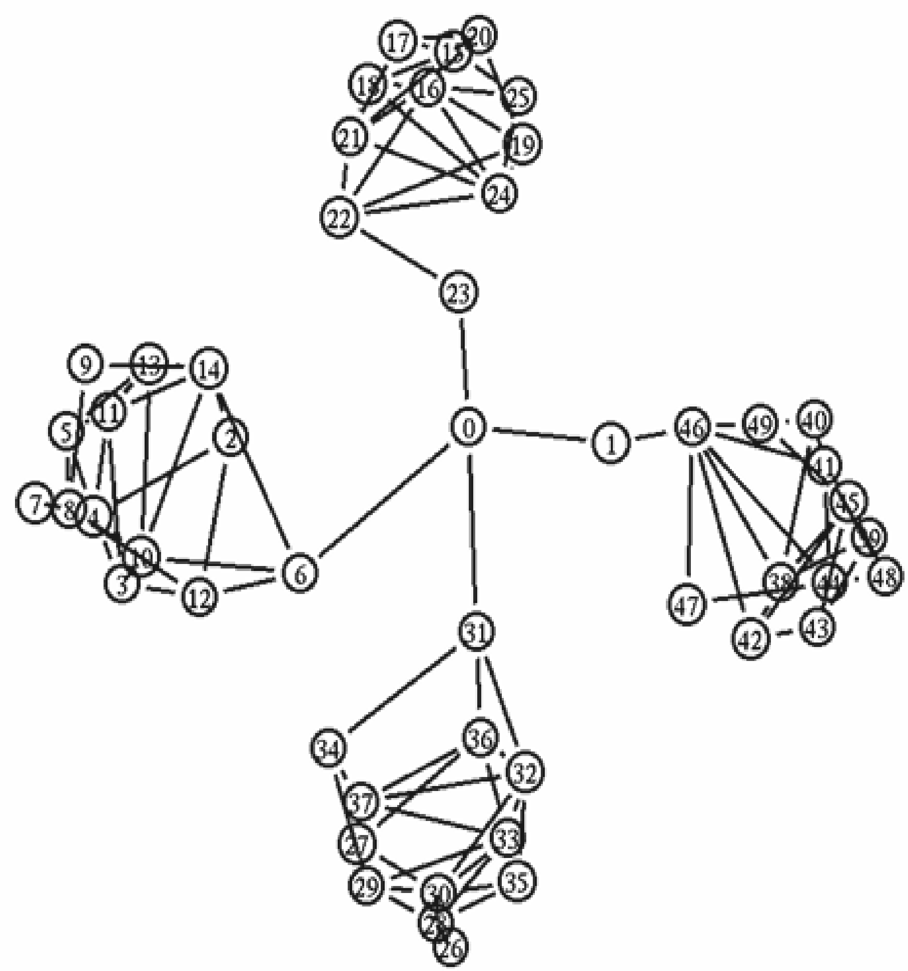


Figure 8. Transit stub topology

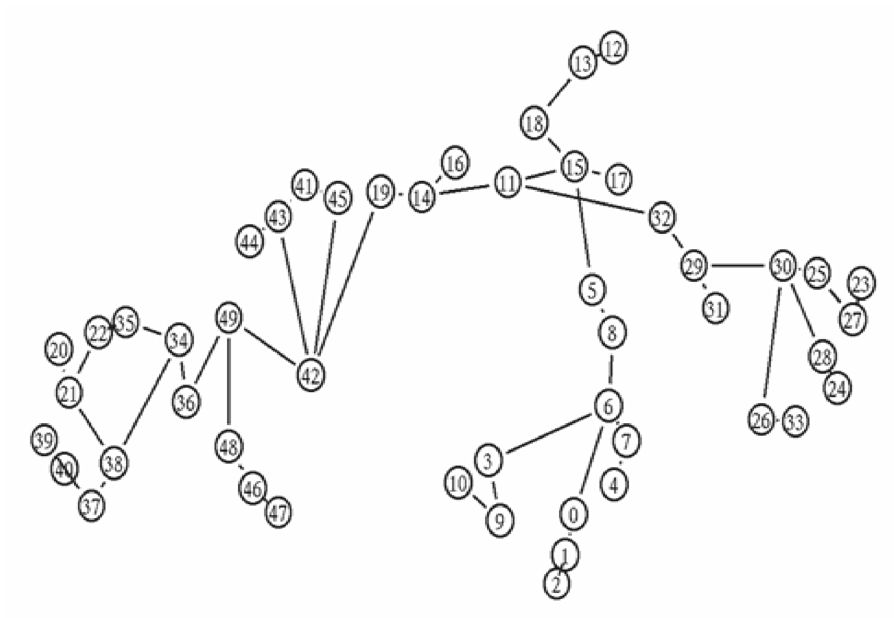


Figure 9. Hierarchical Topology

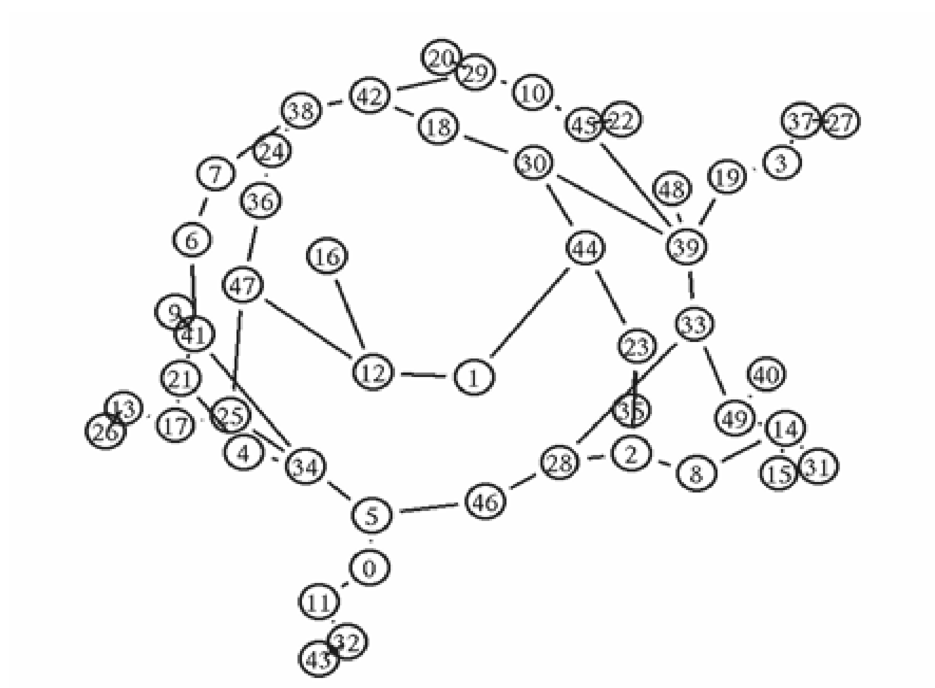


Figure 10. Random Topology

By applying the least square technique for the distances obtained from both TOA and TDOA and at the same time increasing the number of gateway nodes, the following results were obtained. They are summarized in Tables 5 through 9. Other results are shown in the Appendix.

Application of the localization technique yielded the following observations:

- As the number of gateway nodes increases, the location of the unknown node becomes more accurate.
- If the nodes are very close to each other, the location accuracy improves due to minimal processing delays.
- As the distance between the gateway nodes increases, the RTT increases. With high RTT values, processing delays and transmission times become negligible, resulting in more accurate location of the unknown node.
- If the number of gateway nodes is less than 5, then the least square scheme will yield an error of under 3% only when the load is less than 500 kbps.
- The TDOA scheme yields greater location accuracy than the TOA method.

A. *Effect of Network Topology*

Minimum path RTT is related to the topology of the network rather than to the load of the network; hence, it is of most interest to applications and services that are sensitive to network topology. The location of the unknown node is thus highly dependent on network topology, since the RTT is also highly dependent on the network topology. In the transit stub topology, node 46 was unknown. The transit stub topology consisted of 50 nodes with an initial seed of 47. Each transit node was connected to two stub domains, and there were no extra transit-stub edges and no extra stub-stub edges. The reference nodes were 0, 1, 49, 44, 38, 42, and 23. Since node 0 was the transit node, all traffic passing from one domain to another had to pass through this node, concentrating traffic around it. Using this node as a reference influenced the accuracy of the location estimation for the unknown node. Since traffic was high at node 0, the RTT value from the unknown node to this node varied significantly for each traffic load, thus increasing the location error.

Table 5. Error in location for different topologies with 4 reference nodes

Number of Reference Nodes	Topology	Amount of Load	Error Using TOA	Error Using TDOA
4	Transit Stub	No Load	0.069	0.0515
		10%	2.3084	2.828
		25%	3.5906	3.126
		50%	4.7881	4.0
		75%	4.9272	4.90
		90%	6.0997	5.5663
4	Hierarchical	No Load	0.1161	0.101
		10%	3.275	2.9168
		25%	4.340	4.012
		50%	7.250	6.841
		75%	11.551	10.990
		90%	13.2648	12.198
4	Random	No Load	0.0138	0.02
		10%	0.198	0.183
		25%	2.565	1.878
		50%	3.139	3.128
		75%	9.511	6.277
		90%	9.646	9.101

Table 6. Error in Location for different topologies with 5 reference nodes

Number of Reference Nodes	Topology	Amount of Load	Error Using TOA	Error Using TDOA
5	Transit Stub	No Load	0.0688	0.0509
		10%	1.9363	1.3603
		25%	2.9686	2.0613
		50%	3.9645	3.2905
		75%	3.9515	3.7615
		90%	4.8148	4.4615
5	Hierarchical	No Load	0.0821	0.060
		10%	2.9256	2.401
		25%	4.0827	3.317
		50%	6.6358	5.941
		75%	10.642	8.457
		90%	12.280	12.099
5	Random	No Load	0.0368	0.0202
		10%	0.2046	0.170
		25%	0.757	0.640
		50%	0.390	0.829
		75%	2.564	1.1754
		90%	2.969	2.265

Table 7. Error in Location for different topologies with 6 reference nodes

Number of Reference Nodes	Topology	Amount of Load	Error Using TOA	Error Using TDOA
6	Transit Stub	No Load	0.0675	0.0513
		10%	1.741	0.902
		25%	2.629	1.575
		50%	3.23	2.891
		75%	3.405	3.215
		90%	4.128	3.768
6	Hierarchical	No Load	0.0952	0.0528
		10%	3.0766	2.1684
		25%	3.9235	3.1297
		50%	6.280	5.788
		75%	9.657	8.260
		90%	11.159	11.101
6	Random	No Load	0.0447	0.02
		10%	0.117	0.112
		25%	0.913	0.483
		50%	0.529	0.52
		75%	2.851	1.834
		90%	3.3154	2.323

Table 8. Error in Location for different topologies with 7 reference nodes

Number of Reference Nodes	Topology	Amount of Load	Error Using TOA	Error Using TDOA
7	Transit Stub	No Load	0.0843	0.0496
		10%	1.4787	0.8965
		25%	2.2122	1.4389
		50%	2.9876	2.5406
		75%	3.2198	2.3514
		90%	3.4349	2.7729
7	Hierarchical	No Load	0.072	0.046
		10%	2.750	2.0566
		25%	3.8152	2.839
		50%	5.8447	5.3061
		75%	9.0517	5.6786
		90%	10.467	9.4273
7	Random	No Load	0.03	0.03
		10%	0.286	0.211
		25%	0.4062	0.309
		50%	1.220	0.958
		75%	1.7078	1.595
		90%	1.923	1.61

Table 9. Error in Location for different topologies with 8 reference nodes

Number of Reference Nodes	Topology	Amount of Load	Error using TOA	Error using TDOA
8	Transit Stub	No Load	0.07305	0.0486
		10%	1.4604	0.7532
		25%	2.128	1.2862
		50%	2.4588	1.8558
		75%	2.523	2.1283
		90%	2.9079	2.8082
8	Hierarchical	No Load	0.0405	0.02642
		10%	2.2329	1.4381
		25%	2.8849	2.2063
		50%	4.002	3.3067
		75%	6.113	5.0247
		90%	7.1016	5.3185
8	Random	No Load	0.039	0.027
		10%	0.236	0.220
		25%	0.376	0.597
		50%	1.5927	1.304
		75%	1.8414	1.7269
		90%	1.967	1.907

The hierarchical topology uses a trunk node with branches to the other nodes. Although this topology did not use a central node, the traffic was still concentrated on the trunk nodes. The unknown node for this case was 42, and the reference nodes were 19, 44, 11, 14, 48, 49, 43, and

45. Since the traffic had to pass through the higher level nodes when passing from one tier to another, the RTT appears to be high due to the delays incurred by the load. If these nodes are considered the reference nodes for estimating the location of the unknown node, therefore, the error increases due to an increase in RTT values.

The random topology, on the other hand, had no concentrated nodes. The unknown node was 39, and the reference nodes were 1, 44, 19, 48, 45, 30, 40, and 33. When the number of gateway nodes was less than 5, the location error was high for higher network traffic due to increased distance and randomness in the node positions. However, when the number of gateway nodes increased, the error in identifying the location decreased because gateway nodes were closer to the unknown node. The location error was around 1.9% when the load was 90% for eight reference nodes.

B. Effect of Gateway Nodes

The location of the unknown node depends on the locations and number of the reference nodes. When the nodes are closer to the unknown node, with three to four gateway nodes, the location is more accurate than when gateway nodes are far from the unknown node. This greater accuracy is primarily due to lower processing delays, resulting in minimal variations in RTT. Thus, as the number of gateway nodes increases, more information about the unknown node can be gathered and used to improve the location accuracy of the unknown node.

C. Effect of Localization Scheme

As expected, the results show that the TDOA provides greater accuracy than TOA even when the unknown node is far from the reference nodes. Since the TDOA mitigates clock errors by taking the difference of times and since it uses hyperboloid calculations, resulting in a smaller intersecting region, it offers a much greater accuracy than TOA. By contrast, the TOA-based measurement technique yields spherical curves with a larger intersecting region thus generating higher location errors. Besides improving accuracy, the TDOA-based location system also eases implementation since only the receivers (gateway nodes) must be synchronized. Due to range difference measurements, the TDOA scheme avoids transmitter synchronization errors. On the other hand, for TOA-based schemes, the transmitter must be perfectly synchronized with the receivers to estimate the location accurately.

V. CONCLUSIONS

Both TOA and TDOA localization algorithms were applied to computer networks using the least square technique. A simulation analysis indicated that as the number of reference nodes increases, location of the unknown node becomes more accurate. Under increased traffic conditions, the delay and RTT increase, thus degrading location accuracy. With an increased number of hops from the unknown node to the reference node, the RTT also changes significantly. The least square estimation technique yields a 3% error in locating an unknown node. In addition, the TDOA localization method provides better accuracy than the TOA-based scheme. This accuracy can be further enhanced using more reference nodes and an advanced method of interpreting RTT values.

REFERENCES

[1] A. Lakhina, J.W. Byers, M. Crovella and I. Matta, "On the geographic location of internet resources," *IEEE Journal on selected areas in communication*, Vol. 21, No.6, Aug 2003

[2] S. Bhattacharjee, Mostafa H. Ammar, Ellen W. Zegura Viren Shah and Zongming Fei, "Application-layer anycasting," *Proc. of IEEE INFOCOM '97*, Apr. 1997.

[3] M. Ratnasamy, R. Handley, R. Karp and S. Shenker, "Topologically aware overlay construction and server selection," *Proc. of INFOCOM*, New York, NY, June 2002, pp. 1190–1199.

[4] Puneet Sharma, Zhichen Xu, Sujata Banerjee and Sung-Ju Lee, "Estimating network proximity and latency," *ACM SIGCOMM Computer Communication Review*, v.36 n.3, July 2006

[5] P. Francis, S. Jamin, V. Paxson, L. Zhang, D.F. Gryniewicz and Y. Jin, "An architecture for a global internet host distance estimation service," *Proc. of INFOCOM*, New York, NY, Mar. 1999, pp. 210–217.

[6] P. Francis, S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt and L. Zhang, "IDMaps: a global internet host distance estimation service," *IEEE/ACM Transactions on Networking* 9 (5) (2001) 525–540.

[7] W. Stevens, TCP/IP Illustrated, Volume 1: The protocols, Addison-Wesley, 1994.

[8] J. Hightower, B. Brumitt and G. Borriello, "The location stack: a layered model for location in ubiquitous computing," in *WMCSA 2002*, pp. 22–28, IEEE, June 2002.

[9] G. H. Forman and J. Zahorjan, “The challenges of mobile computing,” *IEEE Computer*, 27(4):38–47, Apr. 1994.

[10] M. Weiser, “Some computer science problems in ubiquitous computing.” *Communications of ACM*, July 1993.

[11] B. Karp and H. T. Kung, “GPSR: greedy perimeter stateless routing for wireless networks,” *Proc. of the Sixth International Conference on Mobile Computing and Networking (Mobicom)*, Boston, MA, Aug. 2000.

[12] J. Hightower and G. Borriello, “ Location sensing techniques,” *UW CSE 01-07-01, University of Washington, Department of Computer Science and Engineering*, Seattle, WA, 8 pages, July 2001.

[13] S. Gezici, Z. Tian, G. B. Giannakis, H. Kobayashi, A. F. Molisch, H. Vincent Poor and Z. Sahinoglu, “Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks,” *IEEE Signal Processing Mag*., vol. 22, no. 4, pp.70-84, Jul.2005.

[14] M. Vossiek, L. Wiebking, P. Gulden, J. Wieghardt, C. Hoffmann and P. Heide, “Wireless local positioning,” *IEEE Microwave Mag*., vol. 4, no. 4, pp.88-76, Dec. 2003.

[15] J. G. McNeff, “The global positioning system,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 50, pp.645-652, Mar.2002.

[16] J. Hightower and G. Borriello, “A survey and taxonomy of location system for ubiquitous computing,”, *Technical Report UW-CSE 01-08-03, Univ. of Washington,* Computer Science and Engineering, Aug. 2001.

[17] H. Lim, J. Hou, and C.H.Choi, “Constructing internet coordinate system based on delay measurement”, *IEEE/ACM Transactions on Networking*, Vol. 13, No. 3, June 2005

[18] Y. Shang, J. Ming, and H. Shi, “A new algorithm for relative localization in wireless sensor networks”, *18th International Parallel and Distributed Processing Symposium* 2004

[19] J. Aikat, J. Kaur, F. D. Smith, and K. Jeffay, “Variability in TCP roundtrip times”, *IMC’03*, October 27–29, 2003, Miami Beach, Florida, USA

[20] J.H. Reed, K.J. Krizman, B.D. Woerner, and T.S. Rappaport, “An overview of the challenges and progress in meeting the E-911 requirement for location service”, *IEEE Communications Magazine*, vol. 36, no. 4, pp. 30–37, Apr. 1998.

[21] S. Korkmaz and A.J. van der Veen, “Localization with TOA as a constrained robust stochastic least squares problem”, *IEEE International Conference UWB (ICUWB 2008)*, Hannover, Germany, pp. 197-200,September 2008.

[22] F. Izquierdo, M. Ciurana, F. Barceló, J. Paradells and E. Zola, "Performance evaluation of a TOA-based trilateraion method to locate terminals in WLAN". *Proc. IEEE International Symposium on Wireless Pervasive Computing*, pp. 217-222, Jan. 2006

[23] K.J. Krizmant, T.E. Biedkatt, and T.S. Rappaport, "Wireless position location: fundamentals, implementation strategies, and sources of error", Invited paper, IEEE 1997.

[24] E.K. Bassett, J.P. John, A. Krishnamurthy, D. Wetherall, T. Anderson and Y. Chawathe, "Towards IP geolocation using delay and topology measurements", *IMC'06*, October 25–27, 2006, Rio de Janeiro, Brazil

[25] A. Zivani, S. Fdida, J.F. de Rezende and O.C.M.B Duarte, "Improving the accuracy of measurement-based geographic location of internet hosts", Computer Networks 47, Elsevier, 2005

[26] X. Li and K. Pahlavan, "Comparison of indoor geolocation methods in DSSS and OFDM wireless LAN systems", *IEEE VTS-Fall VTC 2000*. 52nd, Volume 6, 24-28 Sept. 2000, pp. 3015-3020.

[27] V.N. Padmanabhan and L. Subramanian, "An investigation of geographic mapping techniques for internet hosts," *Proc. of the ACM SIGCOMM_2001*, San Diego, CA, USA, 2001

[28] The Network Simulator- ns-2 "http://www.isi.edu/nsnam/ns/"

[29] NS by Example http://nile.wpi.edu/NS/

[30] Stuart A. Golden and Steve S. Bateman, "Sensor measurements for Wi-Fi location with emphasis on time-of-arrival ranging," *IEEE Transactions on Mobile Computing*, VOL. 6, NO. 10, October 2007

[31] J.O. Smith and J. Abel, " The spherical interpolation method for closed-form passive source localization using range difference measurements," *IEEE Journal of Oceanic Engineering*, Vol. OE-12, No. 1, pp. 246-252, Jan. 1987.

[32] B. Friedlander, "A passive localization algorithm and its accuracy analysis," *IEEE Journal of Oceanic Engineering*, Vol. OE-12, No. 1, pp. 234-245, Jan. 1987.

[33] N.T. Spring, "Efficient discovery of network topology and routing policy in the internet," PhD dissertation submitted to University of Washington, 2004.

[34] X. Jin, W.-P.K. Yiu, S.-H.G. Chan and Y. Wang, "Network topology inference based on end-to-end measurements," *IEEE J.Selected Areas in Comm.*, vol. 24, no. 12, pp. 2182-2195, Dec. 2006.

[35] B. Donnet, T. Friedman and M. Crovella, “Improved algorithms for network topology discovery,” *Proc. Sixth Int'l Workshop Passive and Active Network Measurement (PAM '05)*, Mar. 2005.

[36] T.S.E. Ng and H. Zhang, “Predicting internet network distance with coordinates-based approaches,” *Proc. IEEE INFOCOM '02*, pp. 170-179, June 2002.

[37] F. Dabek, R. Cox, F. Kaashoek and R. Morris, “VIVALDI: a decentralized network coordinate system,” *Proc. ACM SIGCOMM '04*, pp. 15-26, Aug. 2004.

[38] R. Siamwalla, R. Sharma, and S. Keshav. (1998, July) “Discovering Internet topology,” Available: http://www.cs.cornell.edu/skeshav/papers.html

II. LOCALIZATION OF NODES IN WIRELESS NETWORKS THROUGH IMPLEMENTATION OF MULTI-INTERFACE MULTICHANNEL(MMCR) PROTOCOL

Hindu Kothapalli, S. Jagannathan, and Maciej Zawodniok
Department of Electrical and Computer Engineering
Missouri University of Science and Technology, Rolla, MO 65401.
hk5y3@mst.edu, sarangap@mst.edu and mjzx9c@mst.edu

ABSTRACT— The design of accurate localization algorithms for wireless ad hoc and sensor networks is challenging due to limited hardware capabilities and the need for cost effective, and low-power processing solutions. This paper presents a means to localize unknown nodes for wireless ad hoc and sensor networks using a multi-interface multichannel routing (MMCR) protocol. This proactive protocol minimizes a novel link cost factor defined by throughput, end-to-end delay, and energy utilization to effectively route the information to the required destination. This protocol balances traffic among available channels on a per-flow basis. It uses the concept of multipoint relay nodes (MPRs) that forward data in the network while minimizing the number of hops and the need for communication.

Localization of the unknown nodes is performed by measuring the received signal strength indicator (RSSI) from an appropriate number of relay nodes during route setup messages. These RSSI values are related to the distance of the unknown node from the relay nodes using a path-loss component. The least squares technique is applied to the distance values to deliver the location. The experimental results using the Generation 4 Smart Sensor Node (G4-SSN) network at Missouri S&T demonstrate the satisfactory performance of the routing protocol and localization scheme.

I. INTRODUCTION

Wireless networks are prone to interferences and channel problems that reduce coverage and capacity, limiting the effectiveness of the system [1]. Communication in such networks is limited to single channel and limited bandwidth [2, 16]. Multiple non interfering channels are available in typical wireless networks. By combining the capacity of these channels, overall bandwidth can be increased and system performance improved. An ad hoc wireless network has a dynamic topology in which data can be relayed by intermediate nodes. Similar to a sensor network, an ad hoc network has limited battery power, transmission range resources [3], and limited bandwidth. All these factors must be considered when designing or implementing a new protocol. On the other hand, a sensor network behaves like an ad hoc network at the cluster-head level; therefore, any protocol for an ad hoc network can be used for sensor networks with certain modifications.

A mobile ad hoc network (MANET) is a group of mobile wireless nodes that form a dynamic network topology with no centralized administration or fixed infrastructure. The mobility of the nodes requires establishing and breaking connections as necessary while maintaining direct communication directly with the nodes within the wireless range of the source. However, the nodes must collaborate to deliver the information between nodes beyond this range. In terms of transmission, each node in a MANET operates in either source mode or router mode. Source nodes generate traffic on the network, whereas routing nodes receive packets and forward them to the intended destination. Figure 1 shows an example of an ad hoc network with various nodes relay the data to the destination in multiple hops.

Most existing routing protocols are limited to a single channel and do not use the overall bandwidth from multiple channels. However, multichannel protocols [4-10] deal with only one QoS parameter, such as throughput, end-to-end delay, or round trip time. Raniwala, Gopalan, and Chiueh [4] proposed a multichannel ad hoc network architecture for wireless mesh networks using centralized channel assignment, bandwidth allocation, and routing algorithms. Wiwatthanasaranrom and Phonphoem [5] introduced a medium access control protocol (MAC) that allows nodes to negotiate channels dynamically, permitting multiple simultaneous communication flows in the same region, with each flow using a different channel.

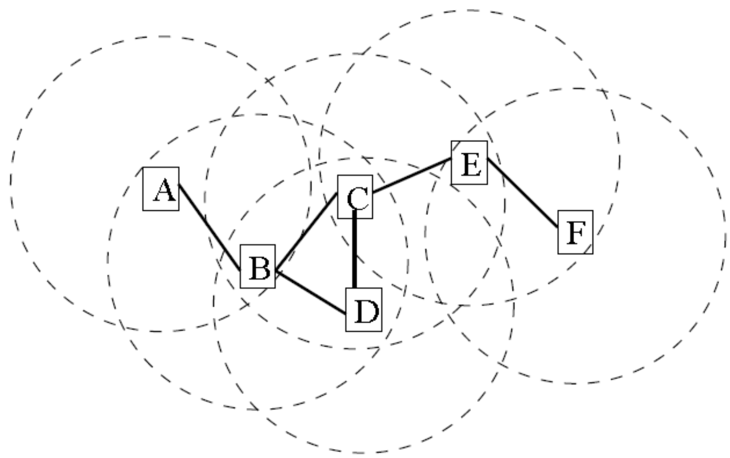


Figure 1. Ad-hoc Multi-hop Network

(Reprint from [28])

Dynamic channel assignment (DCA), proposed by Wu, Lin, Tseng, and Sheu [6], assigns channels dynamically on demand. This protocol assigns one channel for control messages and other channels for data. Each host has two transceivers so that it can listen on the control and data channel simultaneously. Nodes exchange request-to-send (RTS) and clear-to-send (CTS) frames on the control channel, and the data channel is assigned using RTS and CTS messages. This protocol does not require synchronization; however, as the number of available data channels increases, bottleneck in the control channel prevents full use of data channels.

Pathamasuntharam, Das, and Gupta [7] present the primary channel assignment-based MAC (PCAM) protocol, which is based on the use of primary channel assignment with three half-duplex transceivers per node. The primary interface serves as a means for other nodes to contact the node in its primary channel. The secondary transceiver is used mainly for sending data and is not assigned any fixed channel. In addition, a fixed common channel is assigned to the third transceiver to implement broadcast messages. The scheme proposed here requires primary channel discovery instead of channel negotiation since primary channels are pre-assigned.

Gong and Midkiff [8] proposed a family of distributed channel assignment protocols that combine routing with channel assignment using a single transceiver. These protocols employ a cross-layer approach and present an example realization using the AODV [9] routing protocol.

The scheme achieves significantly lower communication, computation, and storage complexity than existing channel assignment schemes, largely due to the combination of channel assignment with routing. Kyasanur and Vaidya [10] have studied the problem of improving the capacity of multichannel wireless networks using a new strategy that does not require modifications to IEEE 802.11.

Several variations of the optimized link state routing (OLSR) [18] have also been developed for the multichannel scenario. Multichannel OLSR (*m*-OLSR), presented by Lee and Midkiff [11] uses modified routing messages with the number of hops as the routing metric and delivers channel information in a fully distributed manner.

These network protocols have been evaluated in simulation; however, only limited experimental results on hardware performance have been reported. Performance of wireless ad hoc network protocols is traditionally evaluated using network simulators such as NS2 [4, 5, 7, 8, 9, 12, 18, 20, 21, 25], OPNET, PARSEC [13, 14], and GloMoSim [14, 19, 22, 23]. Simulations compare the performance of competing protocols under ideal conditions; however, they rarely evaluate the protocol against realistic hardware constraints or dynamic environments with channel uncertainties. Processing capabilities, on-board battery capacity, and sensor interfacing are all constraints that must be weighed in the design of hardware components.

Ad hoc networks are used in many applications including business environments that provide collaborative computing and crisis management service applications, such as disaster recovery, in which the entire communication infrastructure is destroyed and quick re-establishment of communication is crucial. Communication links and network topology vary, requiring frequent retransmissions and rerouting in a wireless ad hoc network.

The IEEE 802.11 PHY specifications define multiple channels and allow the simultaneous, non interfering use of some of these channels. For example, the IEEE 802.11b and IEEE 802.11g PHY standards provide three orthogonal (non overlapping) channels. Twelve orthogonal channels are available in the IEEE 802.11a PHY standard, allowing multiple communications at the same time to improve effective network capacity. The challenge, however, is to allow a single ad hoc network to use the separate channels provided by a physical layer simultaneously and efficiently to increase effective capacity [2, 16]. Several advantages can be expected from the use of multiple channels [12] in wireless ad-hoc networks, including increased throughput, reduced propagation delay, and the availability of additional services using multiple channels. Complete multichannel wireless ad hoc network architecture requires topology discovery, traffic profiling, channel assignment, and routing.

This paper evaluates the multi-interface multichannel routing protocol developed by Anguswamy et al. [25] to provide optimal routing calculations in energy- and delay- dependant environments. Based on hardware implementation, it reports protocol performance in terms of the throughput, end-to-end (E2E) delay, jitter, route setup time, and drop rate. Experimental results indicate that the proposed routing protocol provides the benefits of using multiple channels without modification to the current IEEE 802.11 MAC protocol. Additionally, this paper presents a 8-bit 8051 variant microcontroller-based implementation platform that uses the 802.15.4 RF communication units. The use of this platform provides high-speed processing, interconnectivity with sensors, and a capable radio frequency (RF) communications unit to provide a development platform for ad hoc and sensor networks. The hardware description addresses considerations and limitations that the algorithm and hardware impose on one another. The software description is also discussed in the paper.

The main contributions of this paper include:

- Presentation of a table-driven proactive MMCR protocol that optimizes the route by considering multiple QoS metrics.
- Implementation of the routing protocol in a hardware environment with real-time voice and data.
- Discussion of application constraints related to hardware and software issues.
- Comparison of the performance of this protocol with optimal energy delay subnetwork routing (OEDSR) for various topologies

II. MULTI-INTERFACE MULTI-CHANNEL ROUTING (MMCR) PROTOCOL

The development of the protocol rests on the following assumptions:

- The channel allocation scheme is receiver based.
- The nodes are equipped with multi-radio interface.

One radio is used for incoming data on a dedicated channel, and another is used for outgoing data, switching between channels according to the incoming channel of the next hop node.

Some of the terms used in this report are defined as follows:

N : Set of nodes in the network

s : Source node

d : Destination node

$N(s)$: Set of one-hop neighbors of node s

$N^2(s)$: Set of two-hop neighbors of node s

$MPR(s)$: Selected Multipoint Relay (MPR) set of node s

The routing metric used in this protocol is given by the utilization metric given as:

Utilization metric (U_{s,n_2}^{MPR}) of the link from node s to a two-hop neighbor node n_2 through the chosen MPR (n_1) is given by:

$$U_{s,n_2}^{MPR} = (BF \times EU) / D \tag{1}$$

$$BF = B_A / B_S \tag{2}$$

$$EU = E_A^{n1} / E_{TX}^{n1->n2} \tag{3}$$

where BF is a bandwidth factor between nodes s and MPR (n_1), B_A is an available (free) incoming bandwidth at the MPR (n_1), B_S is an expected/requested outgoing bandwidth at the source node (s), $E.U.$ is a measure of energy utilization between MPR (n_1) and node n_2, E_A^{n1} is available energy at the MPR (n_1) in Joules, $E_{TX}^{n1->n2}$ is energy used to transmit message from n_1 to n_2, and D is an end-to-end delay from node s to node n_1 in seconds.

The bandwidth factor here ensures that there is sufficient available bandwidth for data transfer. A route is selected only if the BF of all links on the path is greater than one. Consequently, only one route is associated with a flow at any given instant, thus guaranteeing service. However, the route may be dynamic with periodic updates of the MPR set. Energy utilization is a measure of energy depletion due to usage, thus improving energy efficiency. The end-to-end delay is one of the QoS metrics for route selection. The utilization factor given in bits per second is a direct measure of the throughput of the link. By optimizing this factor, a high performance can be achieved. The routing scheme is introduced next.

The proposed protocol consists of the several routing phases: neighbor discovery, MPR selection, topology information declaration, and routing table calculation.

A. Neighbor Discovery

Each node in the network transmits HELLO packets to its neighbors. The headers of these HELLO packets include the transmission time. When these packets are received, the receiving node will extract the delay by using the stamped transmission time of the HELLO packet header; however, this process requires that the nodes be time synchronized. The HELLO packets contain the list of its one-hop neighbors and the energy utilization for each of these neighbors. They also contain information about the channels on which the node can receive data and the available bandwidth in that channel. This information is used by the receiving node to calculate the bandwidth factor of the corresponding link. When HELLO packets are received, each node updates this information on available bandwidth, energy factor, and link delay from their neighbors in the neighbor table.

B. MPR Selection

The MPR nodes are selected in such a way that they cover all the two-hop neighbors in the network from the source node. They forward messages to the two-hop neighbors by optimizing the cost factor and ensuring sufficient available bandwidth on each path to support traffic flows. This function ensures that the path through the MPRs optimizes energy consumption, delay and bandwidth utilization. When any of the above conditions fail, then a new node is added to the set of MPRs, thus increasing the bandwidth.

Figure 2 shows the selected MPR nodes in the network for this protocol. They cover all two-hop neighbors in the network.

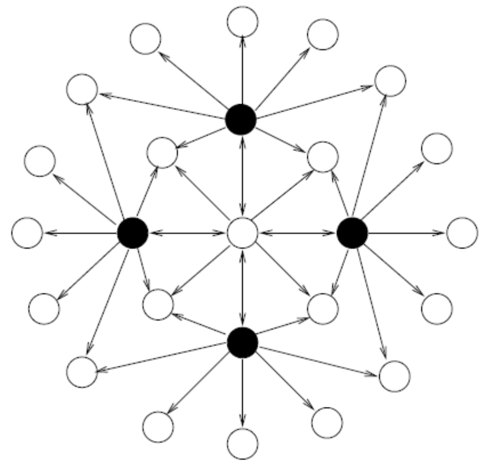


Figure 2. MPR Selection (Nodes filled in black)

C. Topology Information Declaration:

Topology control (TC) messages are transmitted periodically to all other nodes in the network. These messages contain the list of MPR sets with associated cost metric of each. They contain information about the address of the destination, the address of the last hop node to the destination (originator of the TC message), and the cost of the link between the destination and the last hop. Upon receiving these TC messages, each node in the network records the information in the topology table.

D. Routing Table Calculation:

Each node proactively computes the routes to all the destination nodes in the network using the neighbor table and topology table information present at the node. The protocol selects the best route with the lowest cost metric, with the constraint that the bandwidth factor must always be greater than one for all links on the path. The cost factor for a route with k intermediate MPR nodes in the path is given by:

$$C_{s,d} = \sum (C^{n_1}_{s,n_2}, C^{n_2}_{n_1,n_3}, ..., C^{n_{k-1}}_{n_{k-2},n_k}, C^{n_k}_{n_{k-1},d}) \quad (4)$$

where C^{MPR}_{s,n_2} is the cost metric of the link from node s to node n_2 through the chosen MPR (n_1) , and $n_2 \in N^2(s)$ is given by

$$C_{s,n_2}^{MPR} = 1 / U_{s,n_2}^{MPR} \quad . \tag{5}$$

Since maximum performance requires the lowest cost, it is inversely related to the utilization metric.

E. Network Overhead

The header of the HELLO packet for this MMCR protocol is modified to include the transmission time encoded on four bytes. The data portion of the HELLO messages includes an information section for each one-hop neighbor of the sending node. In OLSR [18], only the node IDs are included because that protocol considers only minimum hops in routing calculations. In contrast, the MMCR protocol adds a node's ID, along with information about the delay, available bandwidth at the node, and the energy required to send a packet to the one-hop neighbor. The bandwidth available and the energy information are used to calculate the bandwidth factor and the energy utilization respectively.

A TC message disseminates information about the whole network topology and the selected MPRs. The header remains the same as in OLSR; thus, it includes the number of hops. The data portion of the TC message includes the information for each MPR-selector, i.e., delay and energy for transmitting a packet.

If these HELLO and TC packets are transmitted frequently they increase the overhead and reduce the capacity available for data transmission. If they are transmitted only occasionally, however, they may not detect a one- or two-hop neighbor moving away. Hence, there is a tradeoff between overhead and topology changes.

Additionally, HELLO and TC packets contain information about delay, bandwidth, energy use, and time at which the packet was sent. This information is necessary to calculate the cost factor for each link. When the HELLO packet is received, the node calculates the delay of the packet as the difference between the transmission time stamped in the packet at the source and the received time at the destination. The energy use is the difference between the transmission energy stamped in the packet and the received energy [17].

The TC packets also contain information on bandwidth available at the node, available energy at the node, and delay (other than the number of hops).

F. Multiple Channels over a Single Link

Since a node has more than one radio transceiver to receive data, it can simultaneously exploit multiple non interfering channels. The combined available bandwidth of these channels increases the overall capacity of the link. Moreover, the proposed scheme supports efficient load balancing over these channels to support optimal resource utilization, maximum throughput, and minimum response time.

A node with multiple receiving channels (shown in Figure 3,) when selected as an MPR, is capable of receiving data over multiple channels. The receiving node may already be receiving data from a different source through some channel; hence, the bandwidth available among the receiving channels may be different. The data therefore, should be sent in a balanced mode among the various channels for optimal performance.

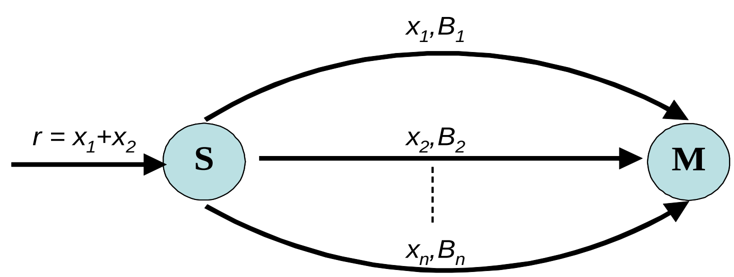


Figure 3. MPR node M has n receiving channels with bandwidths B1, B2 … Bn.

(Reprint from [25])

The following presents a mathematical analysis of load balancing among the various channels.

Bertsekas and Gallager [26] have characterized of optimal routing for directing traffic along paths, which are shortest with respect to some link costs that depend on the flows carried by the links. The cost function for a route can be expressed as

$$\sum_{i,j} C_{ij}(X_{ij}) \tag{1}$$

where Cij is a cost function for link (i,j) as a function of the total Xij over the link

$$X_{ij} = \sum b_p \tag{2}$$

and b_p is a flow through a path containing the link (i,j) .

The problem of identifying the best routing path now reduces to minimizing (6) and (7). According to Bertsekas and Gallager [26], the optimal set of flows (b') is achieved when the traffic is split through the following constraint:

$$\sum_{p \in P_w} [\partial C(b')/\partial b_p](b_p - b') \geq 0 \quad . \tag{3}$$

The cost function in the routing protocol presented here is inversely proportional to the bandwidth factor (BF), which is a function of the flow between the links. Thus, the cost function is obtained by

$$C_n(B.F.) = k / B.F._n = k \times B_S / B_A \tag{4}$$

where k is a constant, $B_S = b$, and $B_A = B - b$ where B is channel capacity.

Consider a node consisting of n receiving channels with bandwidths B_1, B_2 … B_n such that $B_1 > B_2 > \ldots >$ Bn. Let b_1 , b_2 , …, b_n be the bandwidths allocated to each channel by the transmitting node. From equations (8) – (10), an optimal solution is achieved for k available channels when the following condition is satisfied for all $j \in [1, k-1]$:

$$\frac{B_j}{(B_j - b_j)^2} - \sum_{\substack{i=1 \\ i \neq j}}^{k-1} \frac{B_i}{(B_i - b + \sum_{\substack{m=1 \\ m \neq i}}^{k-1} b_i)^2} \leq \frac{1}{B_k} \tag{5}$$

In a homogeneous network with similar physical interfaces for each channel, the constraint (12) becomes equal capacity assignment for all channels. In such cases, the optimal solution is achieved when the link bandwidth is equally allocated among all the available channels on the link.

G. *Implementation in MMCR*

The bandwidth available for each receiving channel at each node is sent via HELLO packets to neighbor nodes. The neighbor node receiving these HELLO packets stores the available bandwidth information for each of these channels. The available bandwidth at each node is the sum of the available bandwidths over all channels. This information is used during MPR selection and the routing process.

Once the link is utilized by the traffic, the load balancing is performed on a per packet basis using the criteria presented above in section F. This approach maximizes the utilization of the link compared to a per flow load balancing in which the packets of a particular flow must be routed via the selected channel or interface. In contrast, the proposed scheme transmits all packets over any of the available channels. Hence, even if the flow data rate exceeds the capacity of a single channel, a packet can be transmitted over multiple channels while meeting the performance criteria.

III. OPTIMALITY ANALYSIS FOR MMCR

This section presents the optimality analysis [25], which shows that the proposed routing protocol is optimal in every scenario. The optimal route is defined as the route with the minimum overall cost indicated in the routing protocol.

Assumption 1: If the one-hop neighbor of a node s has no direct link to at least one of the two-hop neighbors of s, then it is not on the optimal path from s to its two-hop neighbors. However, to reach a two-hop neighbor from s through such a node, the path must go through another one-hop neighbor that has a direct link to the two-hop neighbor.

Corollaries 1 and 2 present the case of destination nodes with no direct link to the source node and at a two-hop distance from it. Corollary 1 is in line with N. Regatte's Optimized Energy-Delay Routing (OEDR) in Ad Hoc Wireless Networks [17].

Corollary 1: The MPR selection based on the utilization metric-based MPR selection provides the optimal route from a node to its two-hop neighbor.

Proof: Case I: When the node d in $N^2(s)$ has only one neighbor from $N(s)$, then that node in $N(s)$ is selected as the MPR node. In this case, there is only one path from the node s to d in $N^2(s)$. Hence, the multipoint relay selection algorithm selects this as the best route between s and the two-hop neighbor d in $N^2(s)$.

Case II: When the node d in $N^2(s)$ has more than one neighbor in $N(s)$, the MPR nodes are selected based on the multipoint relay selection criteria.

Consider a node s whose one-hop neighbors are given by $N(s)$, and a particular node d in $N^2(s)$ with multiple nodes $n_1, n_2, ..., n_k$ $(k > 1)$ belonging to $N(s)$ as its neighbors. Let the cost factor to reach d through *k* one-hop neighbors from s be $C_{s,d}^{m_k}$. According to the MPR selection criteria, the multipoint relay node to cover d from s is selected as the node n_i with a cost factor of

MIN [$C_{s,d}^{m_1}, C_{s,d}^{m_2}, ..., C_{s,d}^{m_{k-1}}, C_{s,d}^{m_k}$].

Hence, the MPR selection criteria yield an optimal route from s to its two-hop neighbors in $N^2(s)$ based on the cost metric.

Corollary 2: The set of MPRs selected for its two-hop neighbors is optimal.

Proof: Let k denote the number of one-hop neighbors and j the number of two-hop neighbors for a source s. Let the optimal set of MPRs be [$m_1, m_2, ..., m_k$] and the optimal set of the cost factor associated with the MPRs be represented as $\left[C_{s,n1}^{m_1}, C_{s,n2}^{m_1}, C_{s,n3}^{m_2}......, C_{s,nj-1}^{m_k}, C_{s,nj}^{m_k}\right]$.

Consider a new one-hop neighbor, m_{k+1}, with a direct link to node n_j ; its cost factor is equal to $C_{s,nj}^{m_{k+1}}$.

If $C_{s,nj}^{m_{k+1}}$ is less than $C_{s,nj}^{m_k}$, then by corollary 1, m_{k+1} becomes a new MPR chosen to reach the node n_j, and it is added to the set of MPRs. Consequently, the cost factor set for the MPRs becomes $\left[C_{s,n1}^{m_1}, C_{s,n2}^{m_1}, C_{s,n3}^{m_2}, \ldots, C_{s,nj-1}^{m_k}, C_{s,nj}^{m_{k+1}}\right]$, which forms a new optimal set of MPRs.

Else by Corollary 1, the m_{k+1} is not chosen as a new MPR, and the set of MPRs remains unchanged since it is already optimal.

Corollary 3 and Theorem 1 address the optimality of route selection through the MPRs. The intermediate nodes are MPRs selected by the previous nodes on the path.

Corollary 3: The intermediate nodes on the optimal path are selected as multipoint relays by the previous nodes on the path.

Proof: A node on the route may not be selected as the MPR by the previous node if it does not provide a connection to that node's two-hop neighbors, or if the node does not meet the MPR selection criteria. The node in $N(s)$ of the previous node s does not provide a connection to any node in $N^2(s)$.

Consider Figure 4 below. Node n_2 only connects to node s's one-hop neighbor n_1. The two possible paths from s to d are $s \to n_1 \to d$ and $s \to n_2 \to n_1 \to d$. According to assumption 1, n_2 is not on the optimal path from s to d .

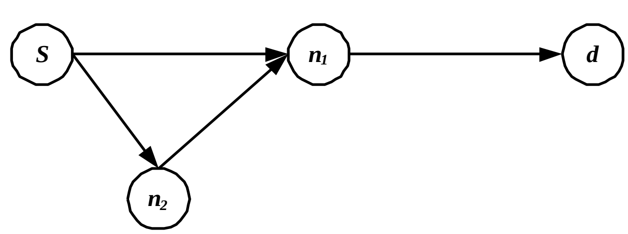


Figure 4. Destination at two-hops

(Reprint from [25])

Based on the Corollaries 1, 2, and 3, the routing protocol always selects the optimal route in terms of the proposed cost metric.

Theorem 1: The multichannel routing protocol selects the optimal route based on the cost metric between any source-destination pair.

Proof: There is an optimal path from source to destination such that all the intermediate nodes on the path are selected as MPRs by their previous nodes on the same path.

Consider the scenario presented in Figure 5. Assume that in an optimal path, $s \rightarrow n_1 \rightarrow n_2 \rightarrow \ldots \rightarrow n_k \rightarrow n_{k+1} \rightarrow \ldots \rightarrow d$, there are nodes in the route, that are not selected as MPRs by their previous nodes. Also, based on the result of corollary 3, assume that for each node on the path, the next node is its one-hop neighbor, and the node two hops away is its two-hop neighbor. For example, n_1 is s's one-hop neighbor, and n_{k+2} is n_k's two-hop neighbor.

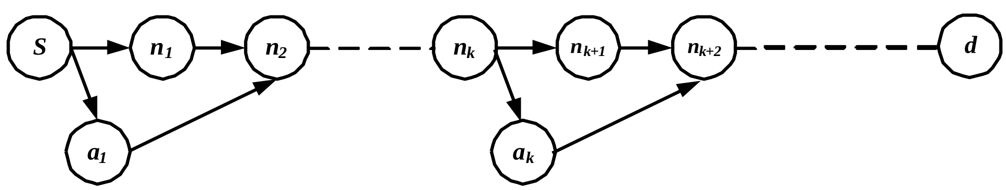


Figure 5. The optimal route scenario between source and destination nodes

(Reprint from [25])

Consider the following two situations:

Suppose that on the optimal route, the first intermediate node n_1 is not selected as an MPR by source s. However, n_2 is the two-hop neighbor of s. Based on the notion that all two-hop neighbors of s must be covered by its MPR set, s must have another neighbor a_1, that is selected as its MPR, and is connected to n_2. According to the MPR selection criteria and corollaries 1 and 2, s selects a_1 instead of n_1 as its MPR since the cost to reach n_2 using a_1 is less than or equal to the cost to reach n_2 using n_1. Since route $s \to n_1 \to n_2 \to \ldots \to d$ is an optimal path and the path $s \to a_1 \to n_2 \to \ldots \to d$ is also an optimal path; the utilization of the path is maximized. Implying that the source's MPR is on the optimal path. Assume that on the optimal route $s \to n_1 \to n_2 \to \ldots \to n_k \to n_{k+1} \to \ldots \to d$, all the nodes on segment $n_1 \to \ldots \to n_k$ are selected as MPRs by their previous node. The next hop node of n_k on the optimal route can now be proved to be n_k 's MPR.

Suppose that n_{k+1} is not n_k 's MPR. Just as in 1, n_{k+2} is the two-hop neighbor of n_k, so n_k must have another neighbor a_k, that is the MPR of n_k and has a connection to n_{k+2}. Again, n_k selects a_k instead of n_{k+1} as the MPR since the cost to reach n_{k+2} using a_k is less than or equal to the cost to reach n_{k+2} using n_{k+1}. Since route $s \to \ldots \to n_k \to n_{k+1} \to n_{k+2} \to \ldots \to d$ is an optimal path, => $s \to \ldots \to n_k \to n_{k+1} \to n_{k+2} \to \ldots \to d$ is also an optimal path. This implies that in an optimal route, the (k+1)th intermediate node is the MPR of the kth intermediate node, and all the intermediate nodes of an optimal path are MPRs of the previous node. Thus, the routing protocol selects the best optimal route based on the cost metric for the route between any source-destination pair.

IV. HARDWARE IMPLEMENTATION DESCRIPTION

An overview of the hardware implementation of the MMCR protocol is presented in this section by discussing the capabilities and limitation of the hardware with regard to the routing protocol.

A. Hardware description and limitations

Implementation of any algorithm through hardware is constrained by the limitations of the hardware. Processing capabilities, on-board battery capacity, interfacing all become constraints that must be weighed during the design of hardware components. Use of specific hardware must be weighed against the precision, speed, and criticality of an algorithm's implementation. Constraints addressed for the implementation of the MMCR were use of low-power, small form-factor, and fast processing hardware. For this protocol, low-power consumption was given the highest priority. In turn, the demand for low power limits the types of processor architectures that can be deployed.

Hence the hardware should be energy conservative; performance oriented and should be of small form factor. Hence the type of processor architecture that can be deployed should be able to satisfy all these demands. Use of Silicon Laboratories 8051 variant family were selected for its ability to provide fast 8-bit processing, low-power consumption, and interface compatibility to peripheral hardware components. Limitations that are incurred through the use of these 8051 variant family are a small memory space and maximum processing speed.

This provides high-speed processing, interconnectivity with the nodes, and a capable RF communications unit to facilitate a development platform for the ad hoc networks. In the next section, a description of the specifications for the hardware implemented nodes will be given.

B. Hardware Architecture

The Generation-4 Smart Sensor Nodes (G4-SSN) [15, 27], as seen in Figure 6 were used to perform the functionality of the sensor nodes in our hardware for the implementation of the MMCR routing protocol. These were originally developed at Missouri S&T and subsequently updated at St Louis University (SLU). These nodes have various abilities for sensing and processing. The former include strain gauges, accelerometers, thermocouples, and general A/D sensing. The latter include analog filtering, CF memory interfacing, and 8-bit data processing at a maximum of 100 MIPS. These nodes have 8K RAM and 128K flash memory that make it a suitable choice for the hardware implementation. Table 1 gives a summary of the specifications of the G4-SSN.

Figure 6. G4-SSN motes.

Table 1. G4-SSN Specifications

I_c @ 3.3V [mA]	Flash Memory [bytes]	RAM [bytes]	ADC Sampling Rate [kHz]	Form-Factor	MIPS
35	128K	8448	100@10/12-bit	100-pin LQFP	100

The hardware architecture used for the MMCR implementation is shown in Figure 7. Synthetic data and voice was sent from the beagle board processor in real time which is read serially through the UART by the 8051 micro-controller present in the G4-SSN mote and processed and sent through the Xbee radio to the required destination through multiple hops based on the routing protocol.

Real time voice spoke through the headset is processed and compressed at the beagle board and sent to the source node serially. The source node packetizes the data adds the routing and the Xbee API header and sends to the required destination. If the destination is in the range of the souce node then the data will be received by the destination node directly. If it is not in the range of the transmitter then the data is relayed through multiple hops till the destination is reached. Figure 8 illustrates the pictural representation of the experimental setup.

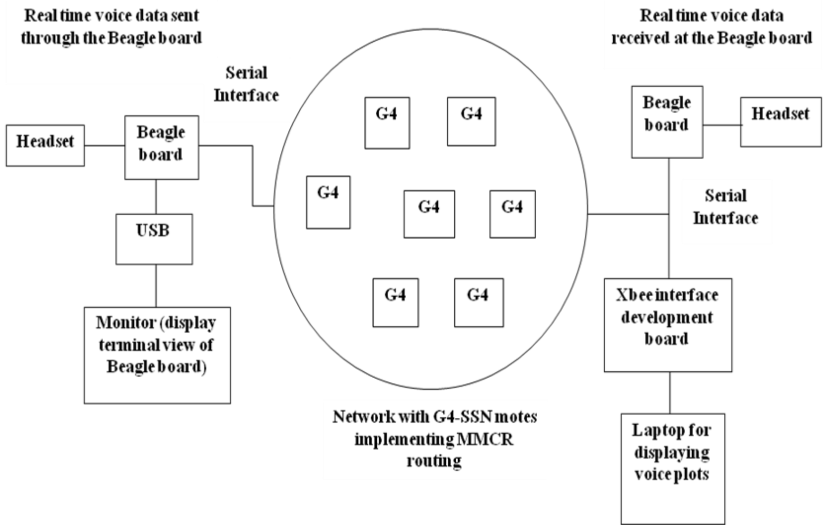


Figure 7. Block diagram of the hardware architecture

The BeagleBoard is an ultra-low cost, high performance, low power OMAP3 based platform designed by BeagleBoard.org community members and sold by Digi-Key. Some of its key features are:

- Includes OMAP3530 processor based on the ARM® Cortex™-A8 core processor to provide a combination of laptop-like performance at handheld power levels in a single chip.
- Has over 1,200 Dhrystone MIPS using the superscalar ARM Cortex-A8 with highly accurate branch prediction and 256KB L2 cache running at up to 600MHz.
- Contains OpenGL© ES 2.0 capable 2D/3D graphics accelerator that is capable of rendering 10 million polygons per second and also a HD video capable TMS320C64x+™ DSP for versatile signal processing at up to 430MHz.
- USB powered.

In the next section, an overview of the software architecture is presented for the MMCR implementation.

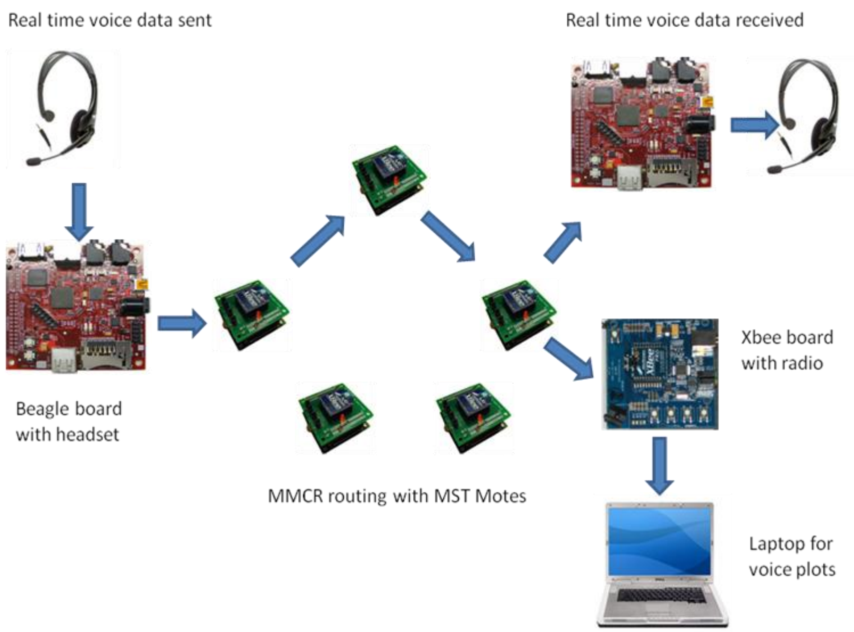


Figure 8. Experimental setup

V. SOFTWARE ARCHITECTURE

The software architecture for the 8051 platform utilized to implement the MMCR protocol on the MST/SLU 8051 motes is shown in Figure 9. A multilevel structure was employed to implement the protocols that allowed separation of software and hardware components. Platform portability and increased ability to modularize the network stack for future additions is accommodated.

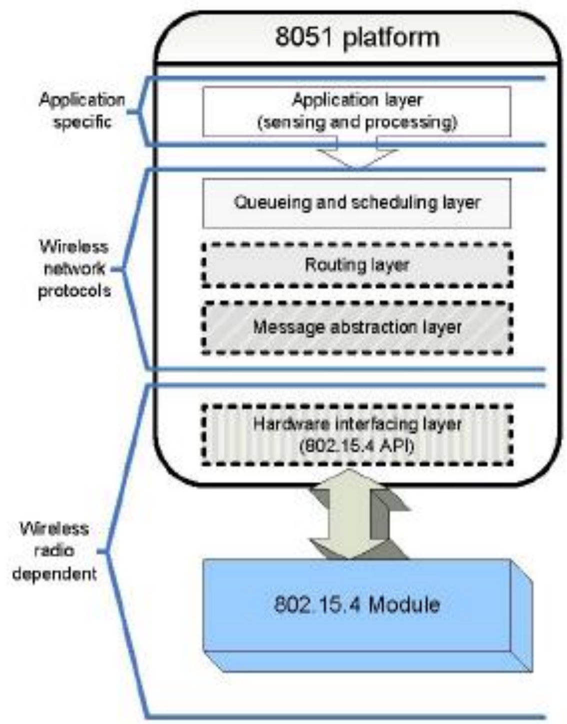


Figure 9. Software architecture

(Reprint from [15])

The three-tier structure provides flexibility to the radio and application design. The wireless radio-dependent components are interfaced with networking layers using a message abstraction layer providing generic access to the physical and link level parameters, for example transmission power level and RSSI indicator. Consequently, cross-layer protocols such as MMCR can be easily implemented.

The main components of the software architecture consists of

- A physical interface between 8051 and 802.15.4 modules—in this set-up a standard serial interface connects the processor with the radio module,
- an abstraction layer—provides generic access to the physical and link layers,
- a routing layer—contains routing implementations,
- a queuing layer—a simple drop-tail queuing policy is employed, and
- an application layer—measurement and processing of sensor data.

VI. ROUTING IMPLEMENTATION

This section describes the routing protocol implementation; includes the packets used by the routing protocol, traffic case handling, and memory handling.

A. Routing Packets

The routing aspects of the MMCR protocol were implemented on the 8051 platform with an 802.15.4 radio module. Five types of messages were considered:

1) BEAM packet

The beam packet is sent from the destination node to all other nodes in the network. This packet is used to attain time synchronization among the nodes.

2) HELLO packet

Each node broadcasts HELLO packets to its neighbors periodically until acknowledgement was received, or until timeout. Based on these packets, all the nodes come to know their one-hop and two-hop neighbors.

3) Acknowledgement (ACK) packet

This packet is sent as a response to the HELLO packet. The HELLO source node receives ACK packet and calculates a transmission delay. The link cost is calculated and temporarily stored to compare it with later responses.

4) Topology Control (TC) packet

When HELLO/ACK timeout has elapsed, the node selects the route using MPR nodes based on the link costs stored in the neighbor table. Thus, these packets contain the MPR nodes and their link costs and sends them to all nodes to indicate the MPR route selection information.

5) SWITCH packet

This packet is broadcasted by the destination node to all other nodes in the network to switch their channel whenever there are many dropped packets due to the interference caused at that channel.

6) DATA packet

The DATA packet conveys application-specific data to the destination node. These data can be synthetic or real-time voice data processed from the Beagle board. They can also be constant bit rate (CBR) traffic generated at the source node itself.

B. Pseudo code for MMCR routing protocol

1) If (hello timeout or start route search)
    - Broadcast HELLO message
    - Start ACK TIMER
    - Return
2) If (Received HELLO)
    - Update one hop neighbor table
    - Send ACK with one hop neighbors
    - Return
3) If (Received ACK)
    - Calculate two hop neighbors with respect to this node based on the one hop neighbors received
    - Calculate link costs for one hop and two hop neighbors and store
4) If (ACK receive timeout elapsed)
    - Select MPR nodes to maximize coverage of two hops
    - Send TC message
    - Return
5) If(Received TC)
    - Store MPR and link cost information
    - Return
6) If (Received BEAM)
    - Update RTC Ticker
    - If (Data to be sent)
        - If (Route available to destination)
            - Send data packet
            - Return
        - Else
            - Start route discovery
            - Broadcast HELLO
            - Return
    - Else
        - Idle Mode
7) For every data packet received, do

- If the packet is destined to monitoring node
    - Accept and move packet to upper layers
    - Return
- Else, packet not destined to monitoring node
    - If monitoring node is the MPR
        - If next hop is the final destination
            - Forward the packet
            - Return
        - Else, next hop is not the final destination
            - Forward the packet to next MPR closest to the destination
            - Return
    - Else, monitoring node is not the MPR
        - Discard the packet
        - Return

C. Traffic Cases

Figures 10 and 11 are block diagrams of the routing control flow information at the transmitter and the receiver respectively. The source node initially broadcasts the HELLO message to all other nodes. The node that receives the HELLO message responds through an acknowledgement containing its one-hop neighbors. This ACK packet when received by the source node stores the two-hop neighbors and selects the MPR nodes based on the link cost. A TC message is then sent indicating the selected MPR nodes and their link costs; hence, packet processing depends on the packet type. Once the packet has been handled, the control flow returns to the idle state and awaits a new packet. The data is sent only after the BEAM packet is received so as to maintain synchronization between the nodes.

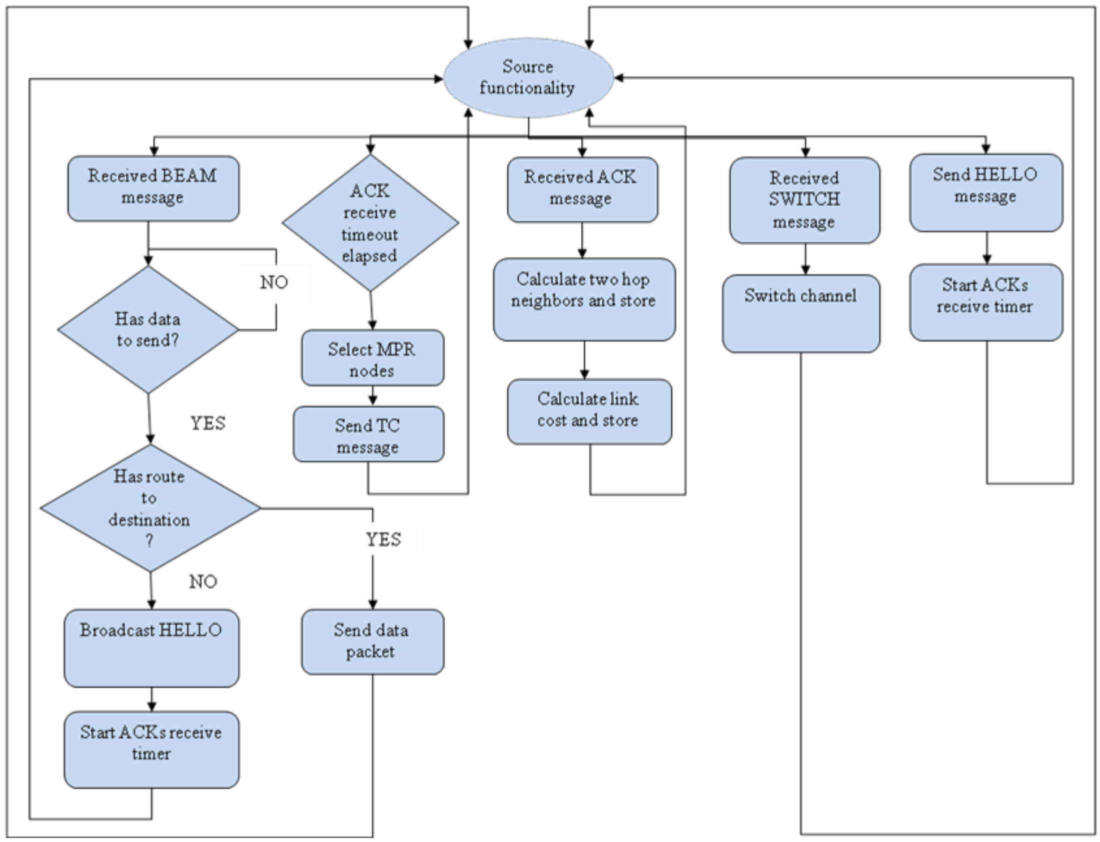


Figure 10. Control flow scheme at transmitter for MMCR routing implementation.

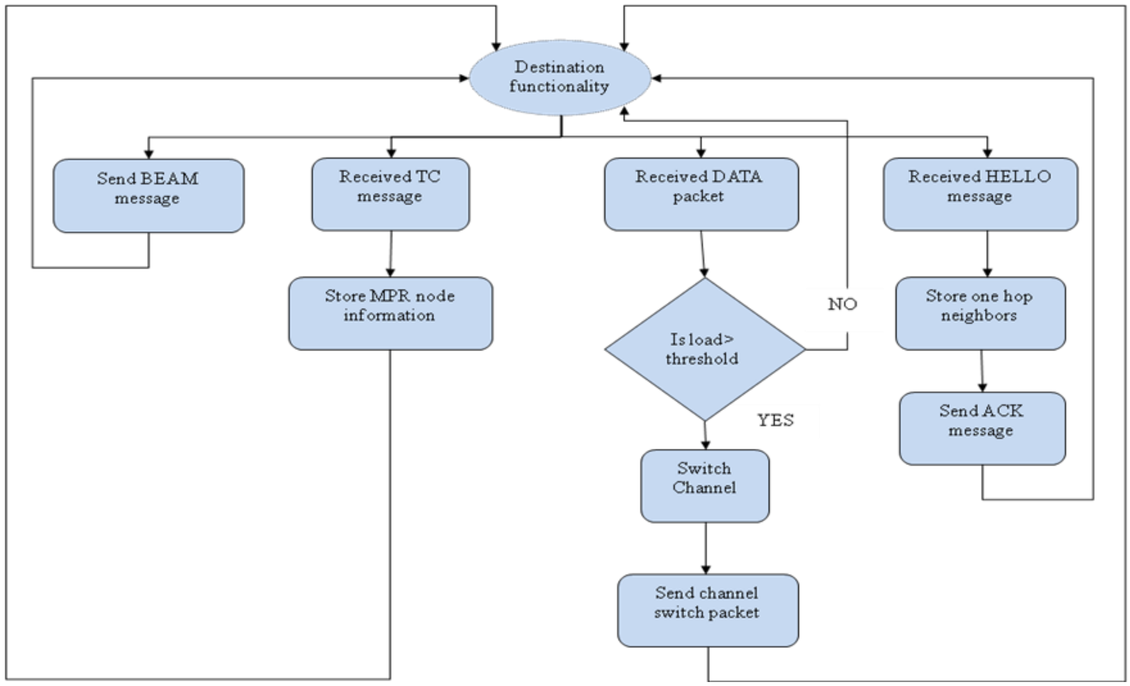


Figure 11. Control flow scheme at receiver for MMCR routing implementation

VII. HARDWARE IMPLEMENTATION LESSONS LEARNED

The MMCR protocol was implemented using a hardware test bed that provided useful information on the deployment issues not encountered during software simulations. These issues include memory limits, network density, RSSI filtering, channel conditions, and other environmental factors.

A. Memory Limitations

Memory limitations are incurred by the hardware. Basic requirements for memory include buffer space for the analog to digital converters (ADCs) and universal asynchronous receiver/transmitters (UARTs), network queues, routing tables with supporting variables, and application-specific buffers and variables. The number of routing table entries depends on the expected number of active nodes. Thus, memory requirements for both the network and the data applications must be considered. For networking applications, memory capacity is most affected by the queuing of the packet flows originating from or passing through a particular node. Hence, an increase in the number of nodes leads to more queuing of the flows, thus becoming more memory-intensive.

For the MMCR protocol, there are several entries for each possible link on the route to calculate the link cost factor. The one-hop and two-hop neighbors must be stored in order to select the MPR nodes. In addition, each entry in the routing table includes the energy, delay, and bandwidth available. Both the bandwidth and the energy are 4-byte values. The addresses stored for each entry in the routing table are 2-byte addresses, one byte is for the node in the direction of the particular destination that is the hop address and the other byte is for the destination node.

The G4-SSN has approximately seven kilobytes for use as queue or application space. Thus it can hold about 70 packets for network queuing.

B. Network Density

The density of the network had a profound impact on the performance of the routing protocol in the implementation. It was measured as the number of nodes per square foot (meter). If the nodes were very close to each other, the node density was very high thus causing severe interference that affected the reliability of the radio channel and increased the route set up times.

High density could also lead to dropped packets or even link failures. Therefore, care must be in distributing active nodes in the network and determining node sleep cycles to limit active nodes.

C. RSSI Filtering

In the routing protocol for each packet, RSSI indicated link reliability. A soft limit was established so that packets with RSSI below an assigned threshold were rejected. These rejected packets were more likely to be corrupted or potentially to pass through a weak link. Therefore, the RSSI filter prevented excessive packet drops and helped to reduce the packet retransmissions for borderline stable links.

A node receiving ACK packets provides an example. If the packet's RSSI value was above the limit, the transmitting node was considered a primary candidate for route selection, and the original MMCR selection criterion applied for such a node. However, when a packet was received with an RSSI below the soft threshold, it was considered only as a fallback choice when no node with an RSSI above the threshold was found.

D. Channel Condition

If the environment was very noisy or had many obstacles, significant interference hindered the overall performance of the protocol. Since the implementation used both 802.15.4 and 802.11x channels sharing the same 2.4 GHz ISM band, they could interfere with each other. Hence, there was a crossover between the 802.15.4 and 802.11x channels. Since this protocol uses multiple channels to transmit the data, when there was significant interference on a particular channel the protocol automatically switched to another channel, thereby improving performance. However, in an environment that is always noisy with poor channel conditions, switching the channel might not make much difference in the performance.

VIII. PERFORMANCE EVALUATION

Experiments for MMCR were performed using a network of G4-SSN's. These experiments were performed mainly to identify practical issues to be considered for protocol redesign and implementation. The performance of the protocol was evaluated in terms of throughput, end-to-end (E2E) delay, route-set-up time, drop rate and jitter. The nodes used 802.15.4 modules transmitting at a 250 kbps RF data rate. The experimental scenario used five nodes placed as shown in Figure 12.

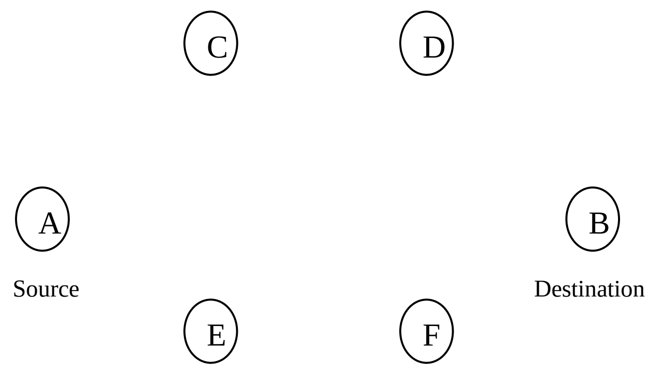


Figure 12. Network schematic

A. Synthetic data

Initially, continuous synthetic data from the beagle board processor was read by the source node through the UART, then processed in its buffers. The route was found using the MMCR routing protocol, and the data was sent from the physical layer of the source node through its MAC layer, reacheing the destination node B through two hops. The route followed was A → C → D → B. The performance plots obtained for the synthetic data is shown in Figures 13 and 14.

Figure 13 displays the transmitted and received data, i.e., before and after routing. Since the average delay for a packet was around 40ms, the received data started from 0.04 seconds. Also, some variations in the received data were observed with respect to the transmitted data because of packet losses due to random channel uncertainties.

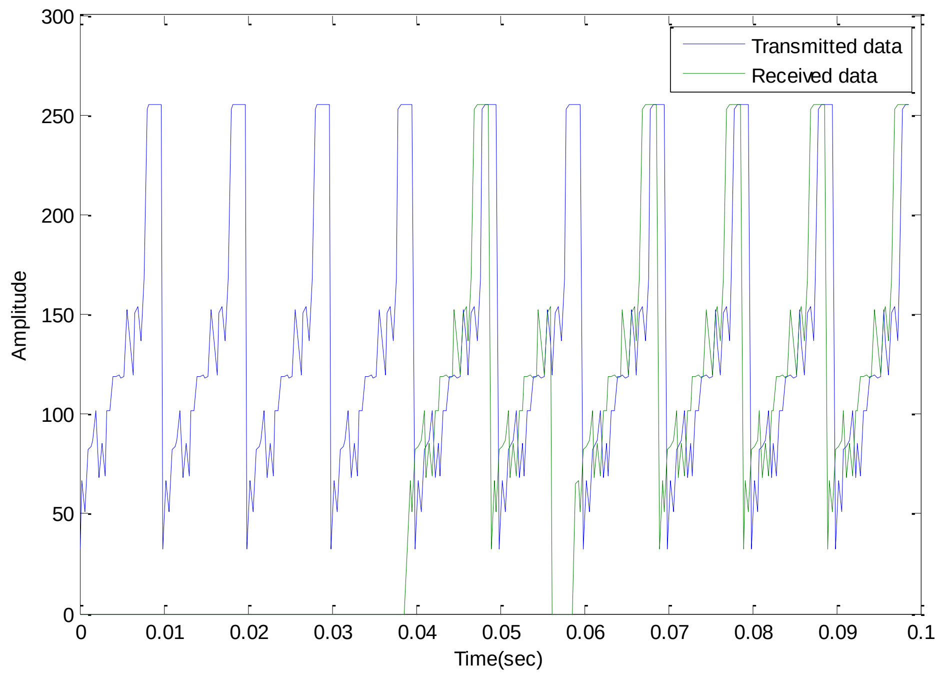


Figure 13. Original data vs. received data

Figure 14 illustrates the various network performance measures for the synthetic data, including throughput, dropped rate, end-to-end delay, and jitter. The throughput was around 15 Kbps, with each packet having a payload of 80 bytes. Some packets were dropped due to channel uncertainties and to the routing set-up that occurs periodically. The end-to-end delay varied from 40 to 50 msec for each packet, with a corresponding jitter of around 5 msec on average. These results are highly satisfactory.

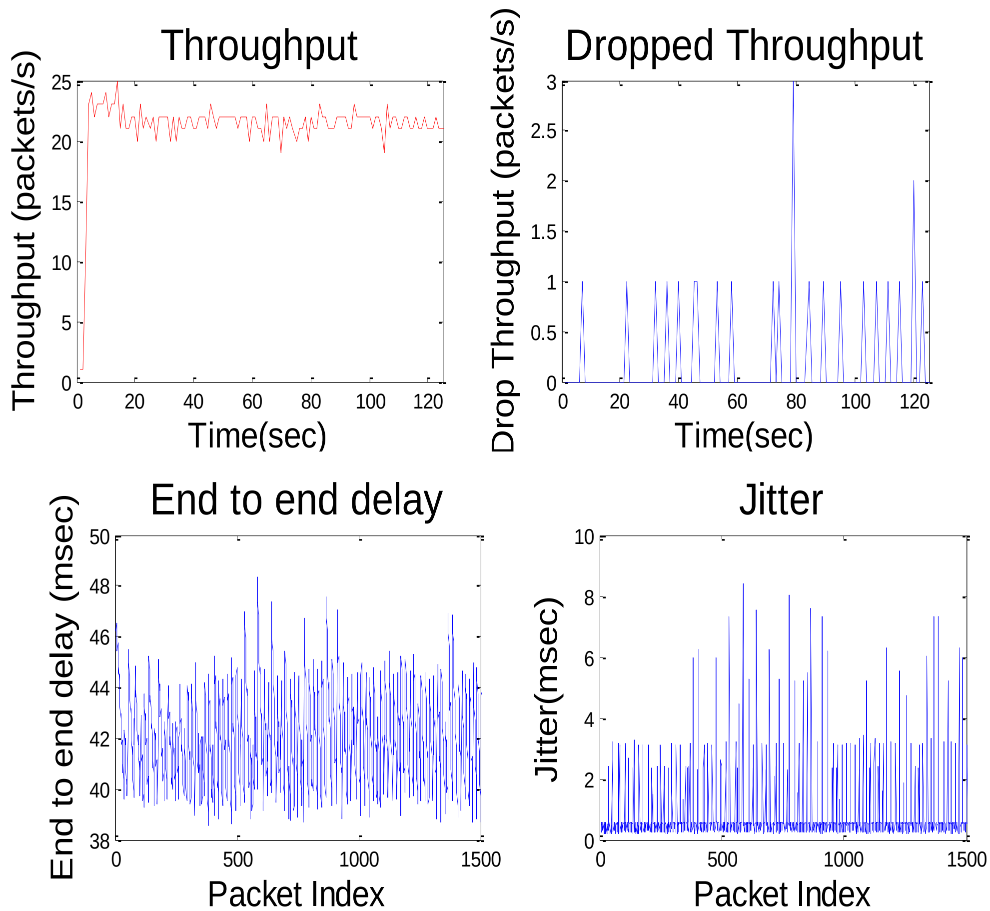


Figure 14. Performance of the MMCR for synthetic data.

B. Voice data

Real-time voice data was then sent from the Beagle board by speaking through the microphone. These data were received by the source node which routes them to the required destination.

Figure 15 shows the original voice and the decoded voice with respect to the sample number. The decoded voice represents the original data with some packet losses.

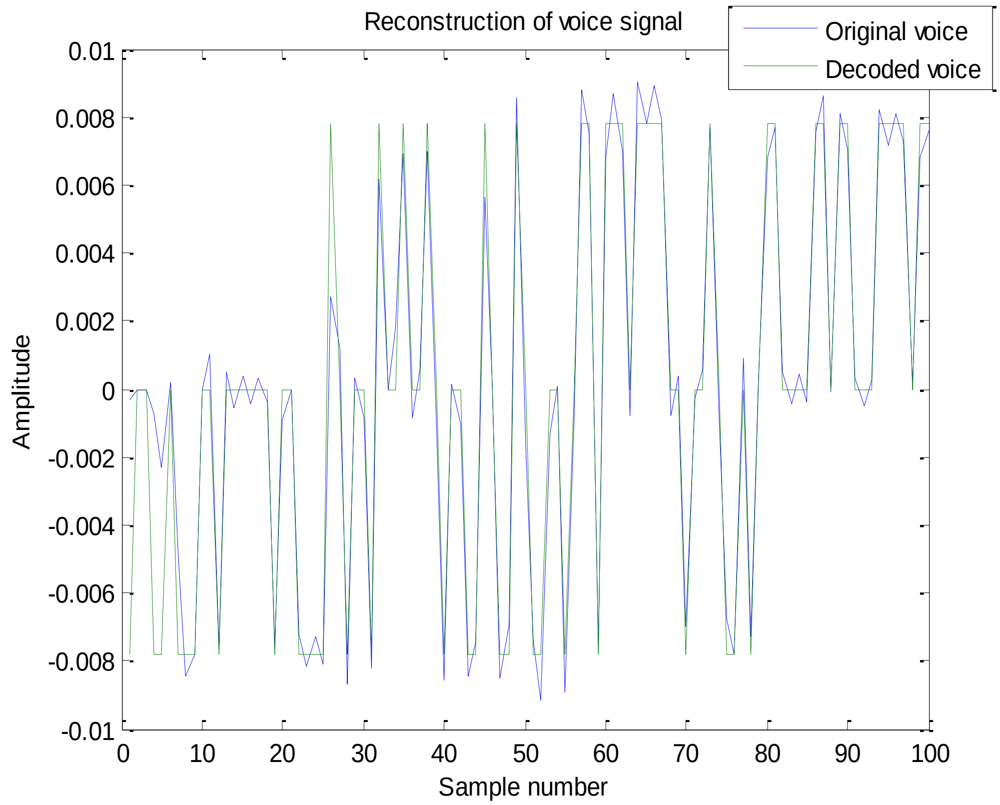


Figure 15. Original vs. received voice

Figure 16 illustrates the various network performance measures for the voice data, including throughput, dropped rate, end-to-end delay, and jitter. The voice data had more dropped packets, it being real time, hence the throughput is comparatively less when compared to the synthetic data. The end-to-end delay is averaged to be 70 ms for each packet.

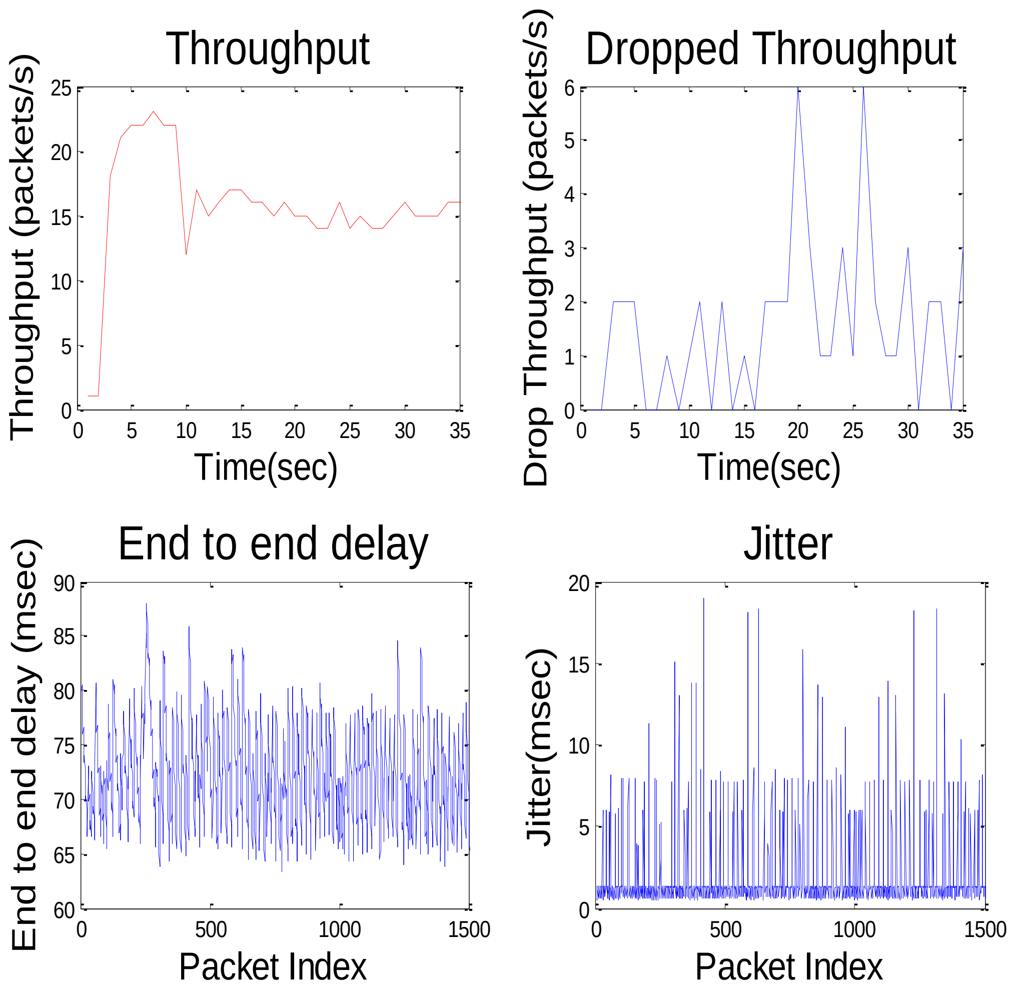


Figure 16. Performance of the MMCR protocol for voice data

Table 2 shows the quantitative results of the performance measures for both synthetic data and voice in terms of the throughput, end to end delay, jitter, dropped packets etc.

Table 2. Performance measures for data and voice

Performance Measures	**Synthetic data**	**Voice data**
Average throughput (kbps)	14.976	10.752
Average End to end delay (msec)	43.93	74.86
Average Jitter(msec)	4.27	9.65
Average dropped throughput(kbps)	0.488	1.644
Average drop rate(packets/sec)	0.763	2.57

The performance measures for both voice and data are calculated using the MATLAB software. The destination radio is connected to the Xbee development board and data received by the radio is sent serially by using a USB cable connected to the PC and the development board. The payload is taken from the data packet received serially from the USB port. Using the timer function in MATLAB, the throughput is calculated as the number of packets received every second with the payload of each packet being 80 bytes.

Since each packet is received with the time stamp the end to end delay is thus calculated using the difference in the time stamp received. Jitter is based on the difference in end to end delay for the successive packets received. The dropped packets are calculated by using the sequence number present in each packet. The sequence number is compared with a temporary variable that is incremented for every packet and the difference will give the number of packets dropped. The dropped packets are added for each one second interval and multiplied by 640 bits to give the number of dropped packets in bits per second.

The voice data is real time and thus sent as and when available. Since the voice is considered to be ftp traffic we can observe that they are some packets getting dropped due to the buffer overflows and random channel uncertainties. Hence the throughput is less for voice when compared to the synthetic data which is being sent at a constant rate. Since the voice which is real time is processed and packetized at the transmitter and sent to the destination the end to end delay is high for the voice when compared to the data.

C. Multi Channel Switching

The MMCR protocol was used to balance traffic on a per-flow basis among the available channels. To test this, the OEDSR protocol [15, 23] was used to create interference with the MMCR routing protocol. The topology is shown in Figure 17.

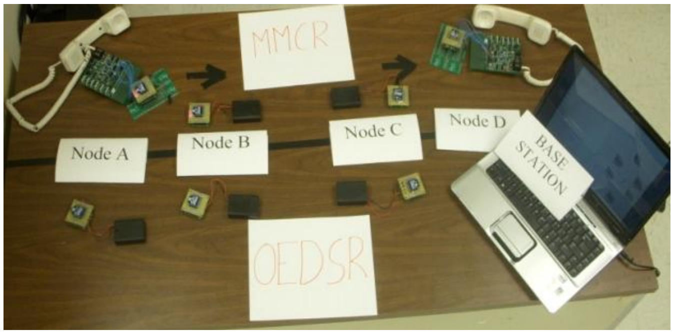
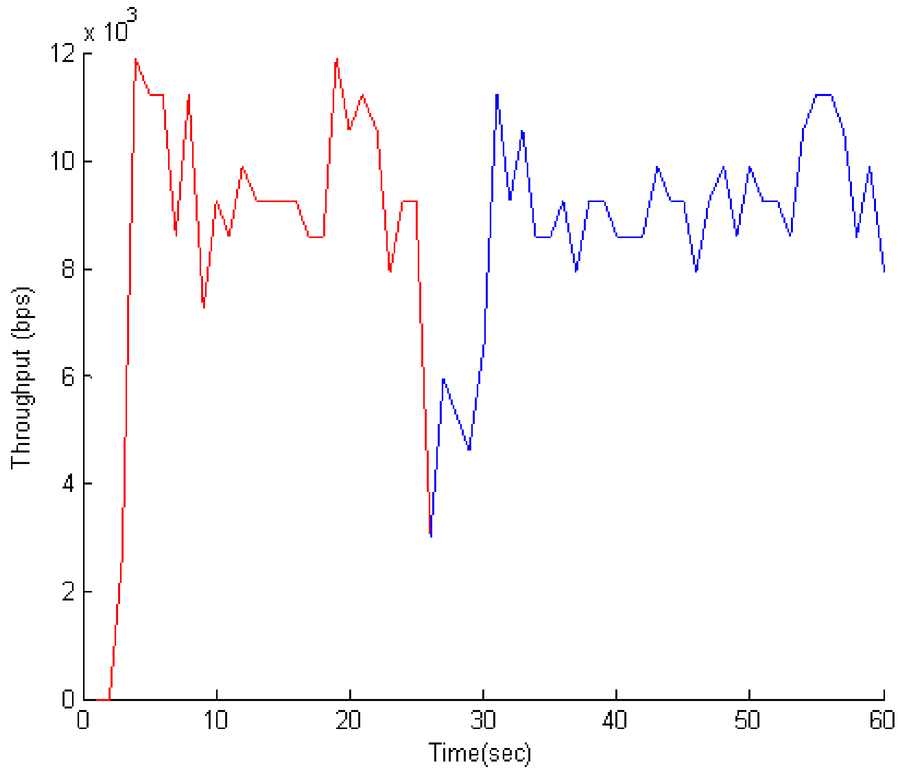


Figure 17. Network setup showing OEDSR and MMCR

Both OEDSR and MMCR protocols send the data on the same channel, in this case channel 17. Since the Xbee radios have 16 non-overlapping direct sequence channels ranging from 0x0B – 0x1A; the default channel was 0x0C (12_d). These channels were used to balance the load, thereby achieving higher throughput. Data was sent at a constant bit rate with a pay load of 80 bytes per packet through the MMCR routing protocol. When the OEDSR began sending its data to the base station, it interfered with the already running MMCR protocol, thereby causing a drop in the throughput. Thus, the MMCR protocol switched to channel 12, as shown in Figure 18. This figure shows how the throughput dropped because of the interference on the channels. The throughput was restored to its original rate after channel switching, demanding that higher throughput can be achieved using various channels.



(Red → Channel 17 Blue → Channel 12)

Figure 18. Channel Switching due to interference

D. Route Setup Time

Figure 19 illustrates the throughput when an active relay node was removed from the network and communication was reestablished again. At 30 seconds, a drop in the throughput was observed when the relay node was removed. Subsequent reestablishment of an alternate route by the MMCR is reflected in the restored throughput.

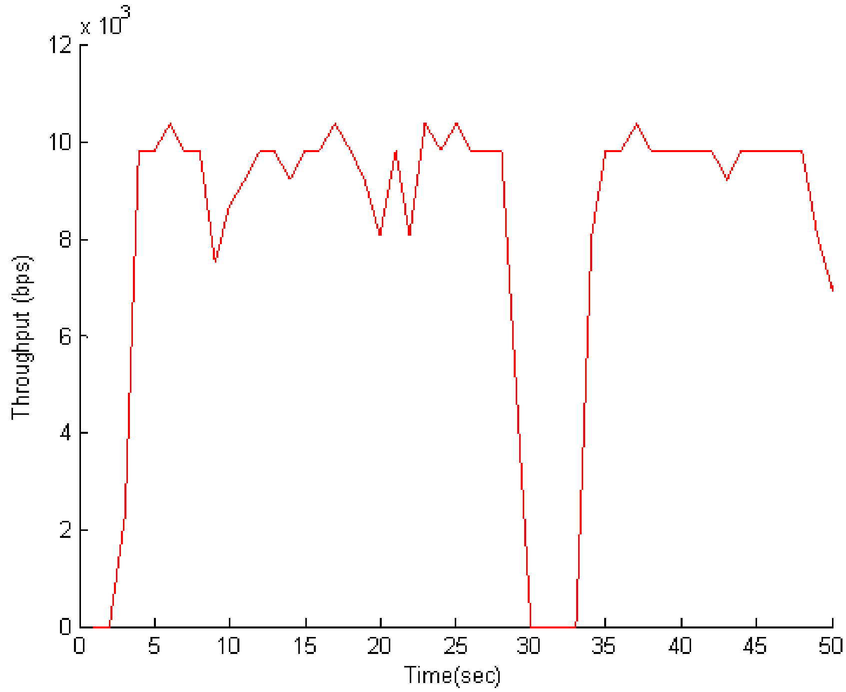


Figure 19. Throughput plot when an active node is removed from the network

Reestablishment of the route took around 2.5 to 3.5 seconds depending on the number of hops and the query time for each hop.

E. Network Overhead Analysis

This work also estimated the overhead for data transmission using the proposed MMCR protocol. The overhead for each data packet with a payload of 80 bytes was 27 bytes, which included 18 bytes of the routing header and 9 bytes of the Xbee API header. The data included in the routing header are summarized in Tables 3 and 4.

Table 3. Fields in the MMCR header of data packet

Start byte	1 byte
Flag byte	1 byte
Mac destination	2 bytes
Mac source	2 bytes
Length of the packet	1 byte
Destination id	1 byte
Source id	1 byte
Module type	1 byte
Module length	1 byte
Report type	1 byte
Sequence number	1 byte
Time stamping	4 bytes
CRC byte after payload	1 byte
Total	**18 bytes**

The Xbee API header includes the following fields:

Table 4. Fields in the xbee API header of data packet

API start byte	1 byte
API length	2 bytes
API id	1 byte
API frame id	1 byte
API destination	2 bytes
API options	1 byte
Checksum byte	1 byte
Total	**9 bytes**

Since the total overhead was 27 bytes, the network overhead was being $\frac{27}{107}\times 100 = 25.2\%$. The network overhead due to only the routing protocol (not including the API header) was $\frac{18}{107}\times 100 = 16.8\%$. Hence, the overhead due to the routing protocol was almost 17% for each packet, which is satisfactory. All these fields are necessary to route the packets, avoid duplicity of packets, and to attain synchronization.

The routing control packets are the HELLO, ACK, TC, BEAM and the SWITCH packets. The HELLO and ACK packets are sent periodically since this is a proactive routing protocol. The HELLO packet is 27 bytes in length including 9 bytes of Xbee API header while the ACK packet contains the list of one hop neighbors and thus the overhead may vary depending on the number of neighbors for that node. The TC packet is sent after the acknowledgement timeout has elapsed to indicate the relay nodes. The BEAM packets are also sent periodically from the destination node to attain time synchronization. Each beam packet is 22 bytes of length which includes 4 bytes of time stamping and also the 9 bytes of Xbee API header. The SWITCH packet is also sent from the destination node whenever the throughput drops and the dropped packets increase due to interference of channels or due to another flow of traffic. Each switch packet is of 18 bytes in length.

IX. PERFORMANCE COMPARISON OF MMCR OVER OEDSR WITH DIFFERENT TOPOLOGIES

Hardware experiments were performed to evaluate the performance of MMCR with various topologies and to compare it with the OEDSR routing protocol. Both central cluster head (CH) topology and grid topology were considered. Experiments were conducted with various numbers of transmission retries for both topologies.

A. Central CH topology

The central CH topology is shown in Figure 20, with the center node as the source node generating traffic at a rate of 3700bps. The mpr nodes selected for this topology are 2, 4 and 6 nodes. Each node used a 802.15.4 module that transmitted at a rate of 250kbps. The nodes' processor interfaces with the 802.15.4 module at 115200 bps. The results obtained forr both MMCR and OEDSR for different number of retries in grid topology are shown in Table 5.

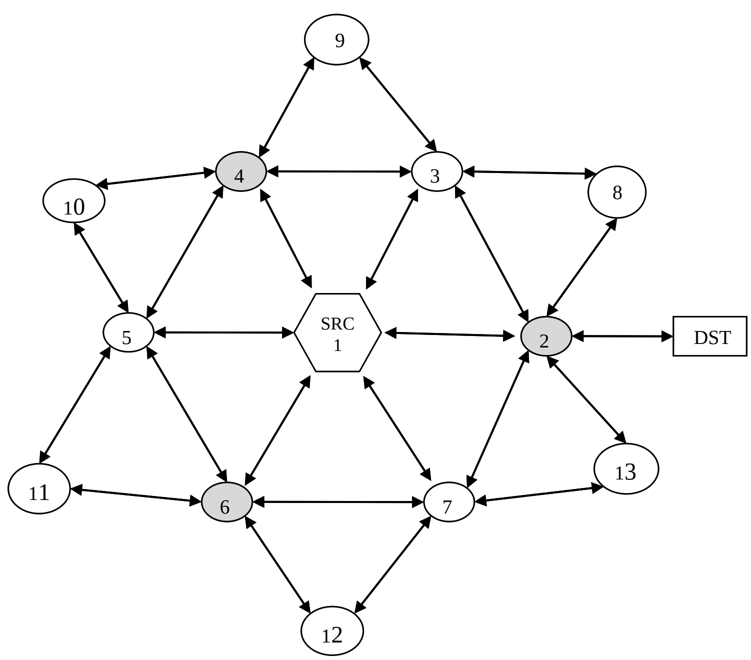


Figure 20. Central CH topology with 14 nodes

Table 5. Comparison of MMCR and OEDSR performance for Central CH topology

Number of retries	6 Retries		9 Retries	
	MMCR	OEDSR	MMCR	OEDSR
Avg. throughput (kbps)	3549.30	3102.00	3418.60	3010.40
Avg. E2E delay(s)	0.188	0.246	0.212	0.274
Avg. drop rate (packets s−1)	0.010	0.103	0.014	0.172
% Energy for routing	1.248	0.427	1.164	0.435
Avg. jitter(s)	0.026	0.043	0.032	0.075
Avg. variance of jitter(s)	0.001	0.003	0.0013	0.006
Avg. route set-up time(s)	2.969	1.550	2.395	1.597

The results clearly indicate that the central CH topology performed much better with MMCR than with OEDSR. This topology demonstrates a worst case scenario for OEDSR protocol since half the network is inactive and only a subnetwork between the cluster head and the base station is active. Differences are especially apparent in the route set-up times and the amount of energy used in routing. The MMCR protocol consumed more energy than the OEDSR in this case because in MMCR all the nodes actively broadcasted HELLO messages to each other, selecting the MPR nodes based on the link costs of the one-hop neighbors. Hence, the number of packets required for routing was higher for MMCR than for OEDSR, and more energy is consumed route data from the source to the destination.

Since MMCR is a table-driven proactive protocol, it periodically updates its neighbor (one hop and two hop) tables and the topology table containing the MPR nodes. Therefore, the route set-up time is more than that required by the OEDSR protocol, which is an on-demand, reactive protocol that selects only relay nodes to reach the base station from the cluster heads. Thus, the OEDSR protocol has shorter average set-up times than the MMCR protocol.

Throughput achieved was higher for MMCR than for OEDSR because it required fewer hops for MMCR than did OEDSR. Also, in case of interference or channel uncertainities, the MMCR protocol switched to another channel, thus reducing interference and restoring throughput.

The maximum number of transmission retries has an impact on protocol performance. The higher the number of retransmissions, the higher the probability of successful transmission, which improves the protocol performance. End-to-end delay and jitter values were lower for MMCR than for OEDSR because MMCR required fewer hops than OEDSR.

B. Grid Topology:

A network of 12 nodes was assembled for testing, as shown in Figure 21.

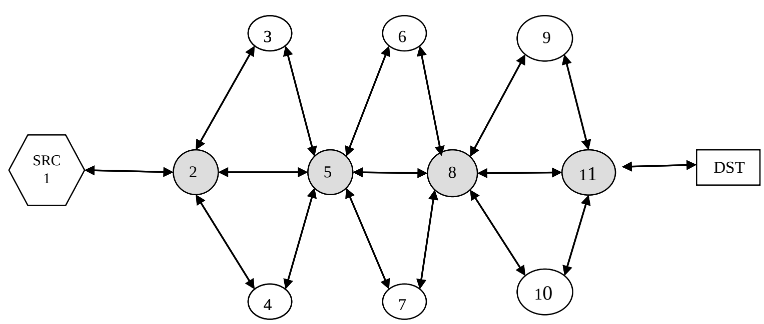


Figure 21. Grid topology with 12 nodes

The grid topology was used to evaluate the protocols in the case of a network using a larger number of hops between the source and the destination. The mpr nodes selected in this case are 2, 5, 8 and 11. Each node used an 802.15.4 module that transmitted at a rate of 250kbps. The source node generated CBR traffic at a rate of 3700bps, and the nodes' processor interfaces with the 802.15.4 module at 115200 bps.

The results for both MMCR and OEDSR for different number of retries in grid topology are shown in Table 6.

Table 6. Comparison of MMCR and OEDSR performance for grid topology

Number of retries	6 Retries		9 Retries	
	MMCR	OEDSR	MMCR	OEDSR
Avg. throughput (kbps)	3265.60	3035.60	3079.40	2811.00
Avg. E2E delay(s)	0.206	0.244	0.237	0.263
Avg. drop rate (packets s−1)	0.024	0.028	0.020	0.022
% Energy for routing	1.450	0.529	0.718	0.515
Avg. jitter(s)	0.034	0.032	0.039	0.043
Avg. variance of jitter(s)	0.0015	0.001	0.0013	0.001
Avg. route set-up time(s)	3.84	2.809	2.419	1.947

In this topology, the average number of hops for both protocols was the same; thus, many of the performance measures, such as the end-to-end delay, jitter, and drop rate, were similar for both the protocols. The results demonstrate that throughput is lower for this topology than for the central CH topology because it requires more hops, which results in more packet losses. However, the throughput is similar for both protocols, with small variations due to random channel uncertainties. Energy consumption is still higher for MMCR than for OEDSR because of its proactive nature and multiple routing phases. Also, MMCR showed a decrease in route set-up time because the number of retries increased due to fewer dropped packets.

The link cost factor for OEDSR was calculated based on delay, distance, and energy remaining at the next hop node. By contrast, the link cost factor for MMCR was calculated based on delay, bandwidth available, and energy utilization. The two protocols had almost the same overhead; however, the MMCR consumed more energy than the OEDSR. On the other hand, due to the use of multiple channels, MMCR performed better in terms of throughput and dropped packets. The interference is reduced by channel switching, even in a noisy or obstacle-rich environment.

X. LOCALIZATION OF THE WIRELESS NODES USING MMCR PROTOCOL BASED ON ZIGBEE 802.15.4 STANDARD

To find the location of an unknown node in a wireless environment, this work used the RSSI values obtained from the multipoint relay nodes. The hardware motes interfaced with the Zigbee radios, which could report the RSSI values for each received packet in dBm units. The seventh byte of every packet received corresponded to the received signal strength. The relay nodes were kept at a distance of one metre from the source node, and RSSI values were taken and averaged. The relay nodes were then kept at different positions, and the signal strength received from the source node to the relay nodes was measured and averaged. Once the location of the relay nodes was determined, the signal strength could be related to the distance, as shown in Figure 22.

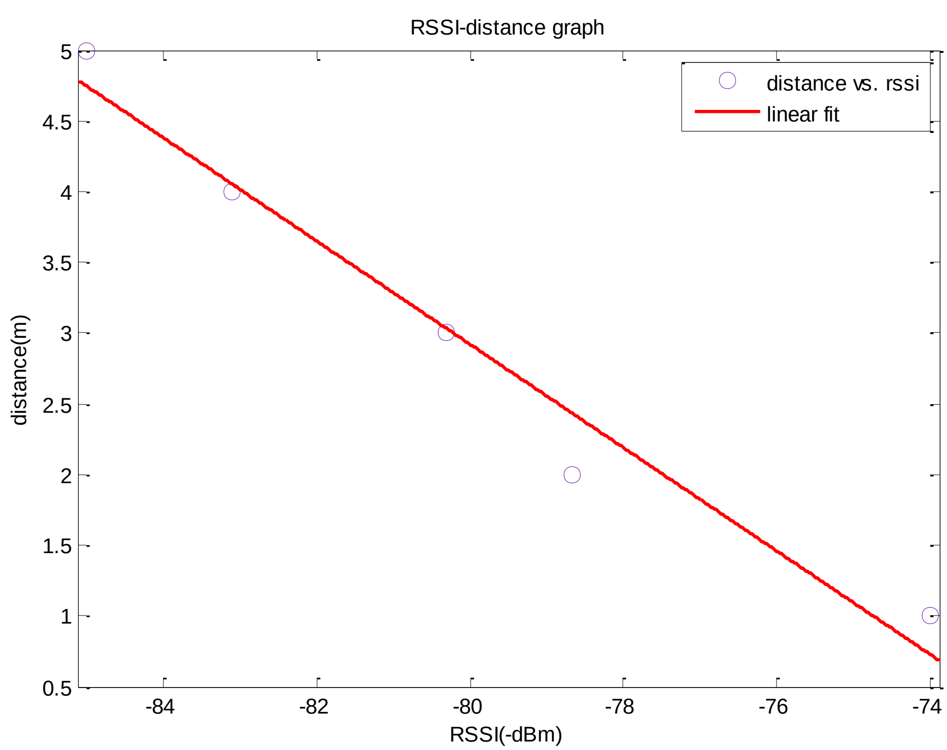


Figure 22. Relationship between communication distance and RSSI value

Based on Figure 22, the slope equation is

$$m = \frac{y_2 - y_1}{x_2 - x_1} \tag{13}$$

The slope of the equation denotes the path loss component and is equal to four for indoor buildings. Hence, equation 13 can be rewritten as

$$-n = \frac{RSSI - RSSI_{1m}}{x_2 - 1} \tag{14}$$

where:

n = path loss coefficient

$RSSI$ = received signal strength from the unknown node to the relay node

$RSSI_{1m}$ = received signal strength when the relay node is placed at a 1m distance

x_2 = radial distance from the relay node to the unknown node.

Hence, the radial distance can be calculated as

$$-x_2 = \frac{RSSI_{1m} - RSSI}{n} + 1 \tag{15}$$

.

Denote the position of the unknown node as (x, y). Let (x_i, y_i) be the positions of the reference nodes where i =1, 2….N.

Once all three distances were determined, the least square technique given by

$$x = (A^T A)^{-1} A^T B$$

was applied, and the approximate location of the unknown node was determined.

The unknown node was kept at various locations with respect to the relay nodes and the RSSI values were taken and averaged.

The relay nodes were kept at

(x_1, y_1) = (0,0), (x_2, y_2) = (1,0), and (x_3, y_3) = (-1,1).

The unknown node (x, y) was kept at (5,3). The RSSI values were measured at these three relay nodes and approximate distances were calculated using equation (15). The distances obtained were 4.825m, 3.868m, and 5.09725m respectively from the unknown node.

Once all three distances were determined, the least square technique given by

$$x = (A^T A)^{-1} A^T B$$

was applied, and the approximate location of the unknown node was determined to be (2.905, -0.350) rending an error of 15%.

The unknown node was placed at different locations and the error obtained in the location by using the RSSI values obtained is given in the Table 7.

Table 7. Error in location using the RSSI technique

Original Location	Observed Location	Error using RSSI (%)
(2,3)	(11.529,0.174)	9.94
(5,8)	(-4.345,2.082)	11.06
(6,4)	(17.580,-3.648)	13.87
(7,5)	(1.058,0.749)	7.30
(8,9)	(0.644,24.731)	17.36
(3,10)	(9.842,-1.663)	13.521

This error appears to be slightly higher than the other schemes like TOA and TDOA. However, RSSI values are naturally available for localization and therefore preferred over other methods.

XI. CONCLUSIONS

A hardware implementation of the MMCR routing protocol was performed for wireless ad hoc and sensor networks. The route selection was based on a metric given by the ratio of the energy available to the end-to-end delay, multiplied by the bandwidth factor, which functions as the link cost factor.

Based on the link factor obtained, the proposed MMCR protocol determined the MPR nodes, which are used to estimate the optimal paths for routing from the source to the destination. This protocol ensures that the selected route is energy efficient, has the shortest end-to-end delay, and has the maximum available bandwidth. Additionally, it maximizes the lifetime of the network by taking energy into account when selecting nodes from a route.

The hardware implementation of this protocol was performed using the G4-SSN motes developed at Missouri S&T. The protocol provided suitable traffic rates and short end-to-end delays. An average throughput of 20 Kbps and an average end-to-end delay of 65 msec were observed for the voice data for a nominal route of 3 hops.

Several issues arose during the hardware implementation. The implementation considered hardware capabilities and limitations, including memory size, processing power, energy consumption, physical size, and interface compatibility with other hardware components. All these issues were explored before the particular protocol was targeted and implemented. The limiting current capabilities of the off-the-shelf radios are some limitations which reduce the overall throughput and increase the end-to-end delay at each hop, when compared to theoretical 802.15.4 capabilities.

REFERENCES

[1] Fundamentals of interference in wireless networks – Spectrum Master MS2711B Application Note. Available at http://www.eu.anritsu.com/files/11410-00302.pdf

[2] P. Gupta and P.R. Kumar, “The capacity of wireless networks,” *IEEE Transactions on Information Theory*, IT-46(2): pp. 388-404, Mar. 2000.

[3] An introduction on ad-hoc networks, WAN Chunfeng presentation, Mita Lab, System Design Department.

[4] A. Raniwala, K. Gopalan and T.-C. Chiueh, “Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks,” *Mobile Computing and Communication Review*, vol. 8, no. 2, pp. 50-65, April 2004.

[5] S. Wiwatthanasaranrom and A. Phonphoem, "Multichannel MAC protocol for ad-hoc wireless networks," in *Proc. National Computer Science and Engineering Conference*, pp. 115-120, Oct. 2003.

[6] S.-L. Wu, C.-Y. Lin, Y.-C. Tseng and J.-P. Sheu, "A new multichannel MAC protocol with on-demand channel assignment for multihop mobile ad hoc networks," in *Proc. International Symposium on Parallel Architectures, Algorithms and Networks*, pp. 232-237, Dec. 2000.

[7] J. S. Pathmasuntharam, A. Das and A. K. Gupta, "Primary channel assignment based MAC (PCAM) - a multi-channel MAC protocol for multi-hop wireless networks," in *Proc. Wireless Communications and Networking Conference* (WCNC), vol. 2, pp. 1110-1115, March 2004.

[8] M. X. Gong and S. F. Midkiff, "Distributed channel assignment protocols: a cross-layer approach," in *Proc. IEEE Wireless Communications and Networking Conference* (WCNC), vol. 4, pp. 2195-2200, March 2005.

[9] C. E. Perkins, E. M. Belding-Royer, and S. R. Das, "Ad hoc on- demand distance vector (AODV) routing," IETF RFC 3561, July 2003. Available at http://www.ietf.org/rfc/rfc3561.txt.

[10] P. Kyasanur and N. Vaidya, "Routing and interface assignment in multi-channel multi-interface wireless networks," in *Proc. IEEE Wireless Communications and Networking Conference* (WCNC), vol. 4, pp. 2051-2056, March 2005.

[11] U. Lee and S. F. Midkiff, "OLSR-MC: a proactive routing protocol for multi-channel wireless ad-hoc networks," *Proc. of the IEEE Wireless Communications and Networking Conference, (WCNC 2006),* vol. 1, pp. 331-336, September, 2006.

[12] J. So and N. H. Vaidya, "A routing protocol for utilizing multiple channels in multi-hop wireless networks with a single transceiver," Technical. Report, University of Illinois at Urbana-Champaign, October 2004.

[13] R. Bagrodia, R. Meywr, M. Takai, Y. Chen, X. Zeng, J. Martin, B. Park and H. Song, "PARSEC: a parallel simulation environment for complex systems," *IEEE Computer*, 31(10):77-85, October 1998.

[14] R.Chandra, C. Fetzer and K. Hogstedt; "Adaptive topology discovery in hybrid wireless networks"; *Proc.of Informatics 1st International Conference on Ad Hoc Networks and Wireless,* vol. 16, pp. 1–16, September 2002.

[15] J. Fonda, M. Zawodniok, S. Jagannathan and S.Watkins, " Development and implementation of optimized energy-delay sub-network routing protocol for wireless sensor networks, *Proc. of the IEEE International Symposium on Intelligent Control*, Munich, Germany, 2006, 119-124.

[16] P. Kyasanur and N. Vaidya, "Capacity of multi-channel wireless networks: impact of number of channels and interfaces," *Proc. of MobiCom*, pp. 43-57, August 2005.

[17] N. Regatte and S. Jagannathan "OEDR: optimized energy-delay routing in ad hoc wireless networks," *Proc. of the World Wireless Congress*, May 2005.

[18] T. Clausen and P. Jacquet, "OLSR: optimized link state routing protocol", IETF MANET Working Group, Internet Draft, draft-ietf-manet-olsr-11.txt, July 2003. Available at http://www.ietf.org/rfc/rfc3626.txt.

[19] About "GloMoSim: Global Mobile Information Systems Simulation Library" http://pcl.cs.ucla.edu/projects/glomosim/

[20] The Network Simultor-ns-2 "http://www.isi.edu/nsnam/ns/"

[21] NS by Example http://nile.wpi.edu/NS/

[22] T. He, J.A. Stankovic, C. Lu, and T. Abdulzaher, "SPEED: a stateless protocol for real-time communication in sensor networks," *Proc. Int. Conf. on Distributed Computing Systems* pp 46–55, 2003.

[23] S. Ratnaraj, S. Jagannathan and V. Rao, "OEDSR: optimal energy delay subnet routing protocol for wireless sensor networks," *ICNSC: Proc. IEEE Conf. on Sensing, Networking, and Control (April)* pp 330–5, 2006.

[24] N. Regatte and S. Jagannathan, "Adaptive and distributed fair scheduling scheme for wireless adhoc networks," *Proc. Word Wireless Congress* pp 101– 6, 2004.

[25] R. Anguswamy, M. Zawodniok and S. Jagannathan, " MMCR: a multi-interface multi-channel routing (mmcr) protocol for wireless ad hoc networks," *International Conference on Wireless Algorithms Systems and Applications*, Dallas, TX, Oct 26-28, 2008.

[26] D. Bertsekas and R. Gallger, Data Networks, New Jersey: Prentice Hall, Inc., 1987, pp. 374-380.

[27] J. Fonda, M. Zawodniok, S. Jagannathan and S.Watkins, "Adaptive distributed fair scheduling and its implementation in wireless sensor networks," *Proc.of the IEEE Conference on Systems, Man and Cybernetics*, Taipei, Taiwan, 2006, 3382-3387.

[28] A. Tonnesen "Implementing and extending the optimal link state routing protocol," a thesis submitted to University of Oslo. Available at http://www.olsr.org/docs/report.ps.

2. CONCLUSIONS AND FUTURE WORK

This work applied localization algorithms to both computer networks and wireless networks, in particular to wireless ad hoc and sensor networks. For wired networks, TOA and TDOA techniques were applied, whereas the wireless scenario depended on RSSI values obtained through the implementation of the MMCR protocol route set-up messages. The least squares technique was used in both networks to estimate accurately the location of the unknown node. The simulation analysis demonstrated that as the number of reference nodes increases, accuracy in locating the unknown node improved.

For wired networks, the delay and the RTT increase as the traffic conditions increase, which affects location accuracy. The TDOA scheme provided much greater accuracy than the TOA-based scheme since it provided a much more concentrated hyperbolic region of the unknown node than did the TOA technique, which yields spherical curves. A 3% error in locating the unknown node was calculated using the least squares technique.

For wireless networks, RSSI values obtained from the gateway nodes to the unknown node were used to estimate the distance of the gateway node from the unknown node. The gateway nodes are the MPR nodes selected by minimizing the available energy and bandwidth through MMCR. By minimizing the link cost factor, an energy efficient route is selected, with the shortest end-to-end delay and the maximum available bandwidth. This protocol was implemented on the G4-SSN motes developed at Missouri S&T by taking into consideration hardware capabilities and limitations, including memory size, processing power, energy consumption, physical size, and interface compatibility with other hardware components. The protocol used multiple channels, thereby providing high data rates and short end-to-end delays with fewer dropped packets. The RSSI values obtained were related to the distance using a path loss component, and the location of the unknown node was estimated using the least squares technique based on the locations of the gateway nodes.

Future work will seek to more accurately determine the location of unknown nodes by using advanced methods of interpreting the RTT and the RSSI values. Accuracy can be improved by finding an optimal value of the path loss component, as we are assuming a constant path loss component of four for indoors in the wireless scenario.

APPENDIX

Transit Stub Topology 50 nodes:

Case 1:

Number of gateway nodes: 3 (n0, n1, n49)

Location of the unknown node n46 = (15, 4)

		No load	10%	25%	50%	75%	90%
TOA	**Least Square Solution**	15.1185, 4.0697	18.4376, 3.5482	20.2497, 3.1358	20.9751, 2.6843	21.6807, 2.2672	22.306, 1.8779
	Error	0.1374	3.4671	5.3203	6.118	6.901	7.6079
TDOA	**Least Square Solution**	15.1196, 4.0492	17.8468, 3.6392	19.1162, 3.1594	19.8782, 2.8486	20.6945, 2.2196	22.012, 1.8961
	Error	0.1293	2.5128	4.2012	5.0122	5.9663	7.3208

Case 2:

Number of gateway nodes: 4 (n0, n1, n49, n44)

Location of the unknown node n46 = **(15, 4)**

		No load	10%	25%	50%	75%	90%
TOA	**Least Square Solution**	14.944, 4.0409	17.22, 3.3671	18.369, 2.7579	19.423, 2.1659	19.36, 1.7047	20.325, 1.0248
	Error	0.069	2.3084	3.5906	4.7881	4.9272	6.0997
TDOA	**Least Square Solution**	15.0175, 4.0485	16.792, 3.5468	18.014, 3.1671	18.8571, 2.9382	19.562, 2.2093	19.9823, 1.5179
	Error	0.0515	2.828	3.126	4.0	4.90	5.5663

Case 3:

Number of gateway nodes: 5 (n0, n1, n49, n44, n41)

Location of the unknown node n46 = **(15, 4)**

		No load	**10%**	**25%**	**50%**	**75%**	**90%**
TOA	**Least Square Solution**	14.9472, 4.0441	16.898, 3.6167	17.848, 3.1623	18.743, 2.6934	18.574, 2.3145	19.295, 1.824
	Error	0.0688	1.9363	2.9686	3.9645	3.9515	4.8148
TDOA	**Least Square Solution**	15.0169, 4.048	16.3264, 3.6983	16.9831, 3.4375	18.2196, 3.3207	18.5068, 2.6394	19.0015, 2.0269
	Error	0.0509	1.3603	2.0613	3.2905	3.7615	4.4615

Case 4:

Number of gateway nodes: 6 (n0, n1, n49, n44, n41, n38)

Location of the unknown node n46 = **(15, 4)**

		No load	**10%**	**25%**	**50%**	**75%**	**90%**
TOA	**Least Square Solution**	14.945, 4.0392	16.741, 3.9723	17.613, 3.7128	18.228, 3.7962	18.277, 3.0738	18.942, 2.7734
	Error	0.0675	1.741	2.629	3.23	3.405	4.128
TDOA	**Least Square Solution**	15.0167, 4.0485	15.9014, 3.9861	16.572, 3.8988	17.884, 3.8039	18.2062, 3.7583	18.6684, 3.1369
	Error	0.0513	0.902	1.575	2.891	3.215	3.768

Case 5:

Number of gateway nodes: 7 (n0, n1, n49, n44, n41, n38, n31)

Location of the unknown node n46 = **(15, 4)**

		No load	10%	25%	50%	75%	90%
TOA	**Least Square Solution**	14.93, 4.047	16.472, 3.8596	17.147, 3.467	17.829, 3.0395	17.887, 2.5745	17.863, 2.1022
	Error	0.0843	1.4787	2.2122	2.9876	3.2198	3.4349
TDOA	**Least Square Solution**	15.016, 4.047	15.8964, 3.9866	16.4371, 3.9275	17.537, 3.8641	17.2561, 3.3374	17.587, 3.0019
	Error	0.0496	0.8965	1.4389	2.5406	2.3514	2.7729

Case 6:

Number of gateway nodes: 8 (n0, n1, n49, n44, n41, n38, n42, n23)

Location of the unknown node n46 = **(15, 4)**

		No load	10%	25%	50%	75%	90%
TOA	**Least Square Solution**	14.949, 4.0523	16.46, 3.9655	17.113, 3.7468	17.399, 3.4608	17.407, 3.2438	17.75, 3.0548
	Error	0.07305	1.4604	2.128	2.4588	2.523	2.9079
TDOA	**Least Square Solution**	15.0151, 4.0462	15.7532, 3.9942	16.2822, 3.8987	16.8386, 3.7479	17.0475, 3.4192	17.6689, 3.1264
	Error	0.0486	0.7532	1.2862	1.8558	2.1283	2.8082

Hierarchical Topology 50 nodes :

Case 1:

Number of gateway nodes: 3 (n19, n44, n11)

Location of the unknown node n42 = (4, 10)

		No load	10%	25%	50%	75%	90%
TOA	**Least Square Solution**	4.0894, 9.9259	6.1071, 7.4918	7.269, 7.1445	10.167, 6.1871	14.637, 5.4954	16.092, 4.5465
	Error	0.1161	3.275	4.340	7.250	11.551	13.2648
TDOA	**Least Square Solution**	4.0881, 9.9496	5.9923, 7.8695	6.8127, 7.139	9.898, 6.533	14.112, 5.695	15.986, 7.732
	Error	0.101	2.9168	4.012	6.841	10.990	12.198

Case 2:

Number of gateway nodes: 4 (n19, n44, n11, n14)

Location of the unknown node n42 = **(4, 10)**

		No load	10%	25%	50%	75%	90%
TOA	**Least Square Solution**	4.107, 9.8961	6.3386, 7.1343	7.4862, 6.7024	10.581, 5.2698	15.451, 3.6916	16.817, 2.7449
	Error	0.149	3.698	4.798	8.104	13.073	14.727
TDOA	**Least Square Solution**	4.0974, 9.9495	6.3348, 7.1343	7.3189, 7.027	10.619, 5.898	16.383, 5.497	16.995, 4.315
	Error	0.1097	3.696	4.4557	7.787	13.1763	14.184

Case 3:

Number of gateway nodes: 5 (n19, n44, n11, n14, n48)

Location of the unknown node n42 = **(4, 10)**

		No load	**10%**	**25%**	**50%**	**75%**	**90%**
TOA	**Least Square Solution**	4.0685, 9.9546	5.9001, 7.7754	7.1129, 7.3583	9.7727, 6.7273	14.002, 6.3645	15.416, 5.4739
	Error	0.0821	2.9256	4.0827	6.6358	10.642	12.280
TDOA	**Least Square Solution**	4.0442, 9.9581	5.464, 8.0961	6.5536, 7.8822	9.2342, 7.1892	12.161, 7.779	15.615, 6.609
	Error	0.060	2.401	3.317	5.941	8.457	12.099

Case 4:

Number of gateway nodes: 6 (n19, n44, n11, n14, n48, n49)

Location of the unknown node n42 = **(4, 10)**

		No load	**10%**	**25%**	**50%**	**75%**	**90%**
TOA	**Least Square Solution**	4.0706, 9.9366	5.9216, 7.5972	6.8982, 7.3552	9.3228, 6.6668	12.977, 6.4384	14.252, 5.5922
	Error	0.0952	3.0766	3.9235	6.280	9.657	11.159
TDOA	**Least Square Solution**	4.0409, 9.9665	5.2481, 8.2268	6.3199, 7.8992	9.1125, 7.2862	12.014, 7.998	14.882, 7.805
	Error	0.0528	2.1684	3.1297	5.788	8.260	11.101

Case 5:

Number of gateway nodes: 7 (n19, n44, n11, n14, n48, n49, n43)

Location of the unknown node n42 = **(4, 10)**

		No load	**10%**	**25%**	**50%**	**75%**	**90%**
TOA	**Least Square Solution**	4.0618, 9.9618	5.7813, 7.9036	6.9297, 7.5561	9.2046, 7.3405	12.757, 7.709	14.022, 6.9778
	Error	0.072	2.750	3.8152	5.8447	9.0517	10.467
TDOA	**Least Square Solution**	4.0368, 9.9723	5.1643, 8.3047	6.0028, 7.9877	8.8648, 7.8812	10.553, 8.812	13.216, 8.015
	Error	0.046	2.0566	2.839	5.3061	5.6786	9.4273

Case 6:

Number of gateway nodes: 8 (n19, n44, n11, n14, n48, n49, n43, n45)

Location of the unknown node n42 = **(4, 10)**

		No load	**10%**	**25%**	**50%**	**75%**	**90%**
TOA	**Least Square Solution**	4.0379, 9.9857	5.4138, 8.2716	6.2688, 8.218	7.8009, 8.7463	10.102, 10.367	11.101, 9.9033
	Error	0.0405	2.2329	2.8849	4.002	6.113	7.1016
TDOA	**Least Square Solution**	4.0226, 9.9863	4.8862, 8.8674	5.6471, 8.532	6.9927, 8.5934	8.985, 9.3694	9.3021, 9.5826
	Error	0.02642	1.4381	2.2063	3.3067	5.0247	5.3185

Random 50 nodes:

Case 1:

Number of gateway nodes: 3 (n1, n44, n19)

Location of the unknown node n39 = (25, 50)

		No load	10%	25%	50%	75%	90%
TOA	**Least Square Solution**	24.968, 50.216	24.758, 52.205	24.471, 54.14	23.14, 61.417	22.155, 66.795	21.761, 73.168
	Error	0.218	2.2182	4.1736	11.567	17.034	23.393
TDOA	**Least Square Solution**	25.046, 50.169	25.383, 48.662	24.919, 52.379	22.669, 59.144	20.013, 62.646	19.346, 68.245
	Error	0.175	1.391	2.38	9.436	13.59	19.10

Case 2:

Number of gateway nodes: 4 (n1, n44, n19, n48)

Location of the unknown node n39 = **(25, 50)**

		No load	10%	25%	50%	75%	90%
TOA	**Least Square Solution**	25.007, 50.012	25.157, 50.122	25.733, 47.541	26.694, 47.357	27.138, 40.732	27.122, 40.59
	Error	0.0138	0.198	2.565	3.139	9.511	9.646
TDOA	**Least Square Solution**	25.02, 50.006	25.183, 49.991	25.645, 48.236	25.932, 46.184	26.787, 43.982	27.065, 41.136
	Error	0.02	0.183	1.878	3.128	6.277	9.101

Case 3:

Number of gateway nodes: 5 (n1, n44, n19, n48, n45)

Location of the unknown node n39 = **(25, 50)**

		No load	**10%**	**25%**	**50%**	**75%**	**90%**
TOA	**Least Square Solution**	25.003, 50.021	25.203, 50.026	24.916, 49.247	25.374, 50.112	23.777, 47.746	23.943, 47.225
	Error	0.0368	0.2046	0.757	0.390	2.564	2.969
TDOA	**Least Square Solution**	25.017, 50.011	25.168, 49.972	25.103, 49.368	24.936, 49.173	24.625, 48.886	24.102, 47.92
	Error	0.0202	0.170	0.640	0.829	1.1754	2.265

Case 4:

Number of gateway nodes: 6 (n1, n44, n19, n48, n45, n30)

Location of the unknown node n39 = **(25, 50)**

		No load	**10%**	**25%**	**50%**	**75%**	**90%**
TOA	**Least Square Solution**	25.004, 50.02	25.024, 50.105	25.217, 49.113	25.517, 50.048	25.116, 47.151	25.152, 46.688
	Error	0.0447	0.117	0.913	0.529	2.851	3.3154
TDOA	**Least Square Solution**	25.016, 50.012	25.112, 49.986	25.167, 49.546	25.363, 49.614	25.585, 48.261	24.318, 47.779
	Error	0.02	0.112	0.483	0.52	1.834	2.323

Case 5:

Number of gateway nodes: 7 (n1, n44, n19, n48, n45, n30, n40)

Location of the unknown node n39 = **(25, 50)**

		No load	**10%**	**25%**	**50%**	**75%**	**90%**
TOA	**Least Square Solution**	24.992, 50.036	25.269, 50.099	24.799, 49.647	24.641, 51.167	23.457, 49.268	23.468, 48.837
	Error	0.03	0.286	0.4062	1.220	1.7078	1.923
TDOA	**Least Square Solution**	25.025, 50.030	24.946, 50.204	24.841, 50.265	24.552, 50.847	23.929, 51.183	23.672, 49.088
	Error	0.03	0.211	0.309	0.958	1.595	1.61

Case 6:

Number of gateway nodes: 8 (n1, n44, n19, n48, n45, n30, n40, n33)

Location of the unknown node n39 = **(25, 50)**

		No load	**10%**	**25%**	**50%**	**75%**	**90%**
TOA	**Least Square Solution**	24.99, 50.038	24.931, 50.226	24.75, 49.719	24.424, 51.485	23.19, 49.661	23.171, 49.274
	Error	0.039	0.236	0.376	1.5927	1.8414	1.967
TDOA	**Least Square Solution**	25.013, 50.024	25.142, 50.169	25.486, 50.347	25.962, 50.881	26.24, 51.202	26.358, 51.339
	Error	0.027	0.220	0.597	1.304	1.7269	1.907

VITA

Hindu Kothapalli was born on July 20, 1986, in Hyderabad, India. Hindu received her high school education in Bharatiya Vidya Bhavans Public School, India. She attended Sri Chaitanya Junior College for her intermediate education from July 2001 to May 2003. She then enrolled in the Jawaharlal Nehru Technological University, India, to receive her Bachelor of Technology (B.Tech.) in electronics and communication engineering in May 2007. She earned a Master of Science degree in electrical engineering in December of 2009 while attending the Missouri University of Science and Technology in Rolla, Missouri. She is a member of IEEE.