UNIVERSITY OF ONTARIO INSTIUTE OF TECHNOLOGY

OLSR-BASED NETWORK DISCOVERY IN SITUATIONAL AWARENESS SYSTEM FOR TACTICAL MANETs

By

Z. M. FAIZUL ISLAM

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASc IN ELECTRICAL AND COMPUTER ENGINEERING

FACULTY OF ENGINEERING AND APPLIED SCIENCE

OSHAWA, ONTARIO

JANUARY 2012

Abstract

In this thesis, we propose a high level design for connectivity visualization of OLSR-based MANET topology based on local topology databases available in an OLSR node. Two different scenarios are considered: a central (full view) topology from a command and control location, or a nodal (partial) view from an ad-hoc node. A simulation-based analysis is conducted to calculate total number of active links at a particular time in full and nodal topology views. Also the error rate of network topology discovery based on total undiscovered link both mobile and static scenario is considered and reported. We also come up with an analytical model to analyse the network bandwidth and overhead of using TC, HELLO and custom NIM message to evaluate the performance of centralized visualization to build full map of the network with respect to situational awareness system. This thesis also presents a multi-node, 2-dimensional, distributed technique for coarse (approximate) localization of the nodes in a tactical mobile ad hoc network. The objective of this work is to provide coarse localization information based on layer-3 connectivity information and a few anchor nodes or landmarks, and without using traditional methods such as signal strength, Time of Arrival (ToA) or distance information. We propose a localization algorithm based on a Force-directed method that will allow us to estimate the approximate location of each node based on network topology information from a local OLSR database. We assume the majority of nodes are not equipped with GPS and thus do not have their exact location information. In our proposed approach we make use of the possible existence of known landmarks as reference points to enhance the accuracy of localization.

Acknowledgements

I wish to express my appreciation to my supervisor Asst. Prof. Dr. Shahram S. Heydari for his expert guidance and valuable feedback in the process of completing this thesis. I am likewise grateful to Dr. Anand Dersingh, Md. Mustafizur Rahman, Mitchell Romanuik and Abdullah Alshehri for their support and feedback during the thesis.

I am much obliged to my family and love of my love for encouraging and supporting me throughout my studies, also for believing in me and teaching me to believe in myself.

Finally, I wish to express further gratitude to all of those who supported me in any respect during the completion of the thesis.

Dedication

To my father MD. Abdul Mannan,

and my mother Fazila Khatun.

Table of Contents

List of Tables

List of Figures and Illustrations

Chapter One: Introduction

1.1 **Background**

MANET (Mobile Ad-hoc Network) is a network of nodes, which are connected by wireless links. There is no infrastructure and central command for communicating and controlling those nodes. It is a self-configuring network: the wireless nodes create the network among them. The nodes are always moving and they are working often as routers at the same time [1].

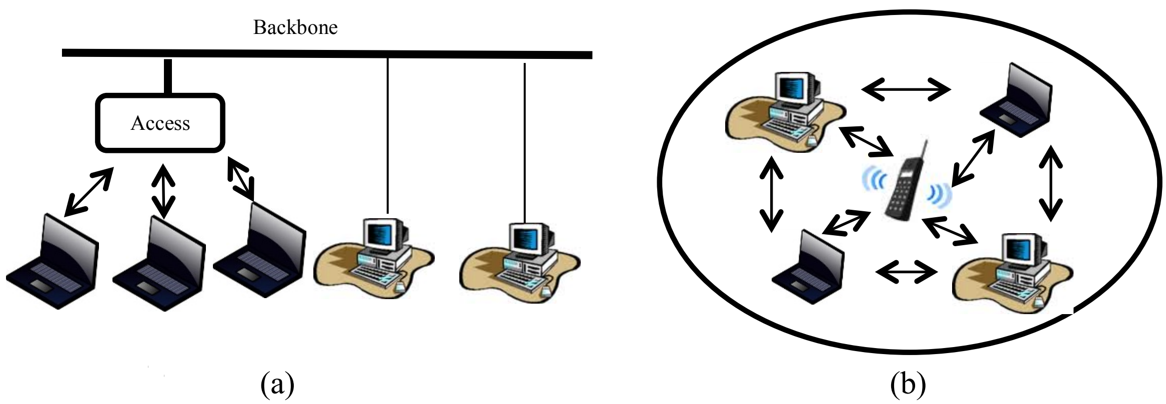


Figure 1-1: Network with (a) infrastructure and (b) without infrastructure.

MANET has the following characteristics:

- An autonomous system of nodes connected by the wireless network.
- All nodes use the same channel.
- Peer-to-Peer connectivity.
- Multi-hop forwarding to ensure network connectivity.
- Nodes are free to move arbitrarily with different speed; thus the network topology may change randomly and at unpredictable times.

- Energy constrained operation: Some or many nodes in an ad hoc network may rely on battery or other exhaustible means for their energy.
- Limited bandwidth.
- Security Threats: Mobile wireless networks are generally more prone to physical security threats than fixed-cable nets. The Increased possibility of eavesdropping and spoofing, and minimization of denial-of-service type attacks should be carefully considered.
- Distributed operation.

The scope of using MANET is increasing day by day with an increase in the use of portable devices and for the set of applications, which are diverse and large in range. It can be created easily and also nodes can be removed and added without hampering the existing network. The application of MANET is widespread. This includes:

- *Military Battlefield:* Now a day's all the military equipment contains the high tech component. It is also tough and quite impossible to create an infrastructure based network in a battle field due to the mobility of soldiers and vehicles on the field. But using ad-hoc network it is possible to maintain an information network among soldiers, vehicles and military headquarters.
- *Disasters Areas:* It is an important application for MANET. This network can be used for emergency or rescue operation in disasters i.e. flood, fire or earthquake. All the emergency vehicles, equipment and people working in a rescue operation can communicate without any existing network infrastructure.
- *Local Level:* In a conference center or in a classroom an instant and multimedia network can be created within the Laptop or PDAs using this ad hoc network.

Also all the households' device can exchange information among themselves using MANET [2].

- *Personal Area Network (PAN):* Intercommunication between different mobile devices can make easier using short range MANET. In the future pervasive computing context PAN is a promising application field of MANET [2].

There are some networking parameters that must be taken into account for calculating the performance of a mobile ad-hoc network. That includes [3]:

- Network size: The size of the network will be calculated according to the number of nodes.
- Network connectivity: The average number of neighbor of a node.
- The topological rate of change: The changing speed of topology for the network.
- Link Capacity.
- Fraction of Unidirectional Link: If there is any unidirectional link present in the network how does a routing protocol work as a function of this link?
- Traffic patterns: Adaptation rate of traffic protocols to non-uniform or busty traffic patterns.
- Mobility of the nodes.

Situational Awareness (SA) is a mission critical networking application in TMANET as it helps the decision maker (e.g., command and control center) to be aware of what is happening in the vicinity [4]. For various fast-moving, life-critical situations such as military combat, fires, disaster relief, it is very important for the command unit to be aware of the dynamic situation for the successful response and usually depends on robust positioning capabilities and effective communication. Thus for a better situational awareness system it is necessary to have location and connectivity information about the nodes within the network. An important task of such system is to create a network Common Operating Picture (COP) to deliver a view of the status of network components in real time, including parameters such as the logical view of interconnectivity and dependencies, logical view of assets and their configuration, geographical information of the assets, and potential threats [5]. This task is particularly difficult in tactical MANETs because of the harsh environment (noise, link instability, and hostile nodes), dynamic topology due to node mobility, limited bandwidth due to wireless communication, limited energy resources for battery powered devices, and limited security against eavesdropping [6].

In developing topology discovery algorithms based on routing information, the following considerations must be taken into account:

- *Active vs. Passive*: Situational Awareness systems can use an active approach to network discovery, e.g. by sending queries to network nodes; or take a passive approach by merely listening to the messages, in particular routing updates and Link State Advertisements (LSA) that are exchanged between network nodes.
- *Proactive vs. Reactive, Distance Vector vs. Link State Routing Protocols*: MANET routing schemes are often categorized as proactive or reactive, depending on whether the routing is performed using a pre-calculated routing table, or on-demand.
- *Centralized vs. Nodal Visualization*: By nodal visualization we refer to the viewpoint of an individual node about the whole network, based on the routing messages received by this node. Depending on the routing protocol, this view could

be partial or complete at different time instances. In centralized visualization we assume that the monitoring node has access to all messages in the network and/or all nodal routing and topology databases. Obviously a centralized network discovery converges faster than nodal network discovery; however it may require active participation, i.e. sending queries to the nodes to collect status information.

Localization of a node can be defined as a particular course of action intended to determine the position of the node based on a coordinate system that can be either real or virtual [7]. Localization is a key feature in Situational Awareness Systems, which provide a real-time Common Operating Picture (COP) of the network, including the status, location, connectivity and other parameters of all network components for a command and control center. Situational Awareness Systems are particularly useful in Mobile Ad-hoc networks (MANET) with application in vehicular, tactical or ambient communication [8]. Such networks share certain characteristics such as low bandwidth, limited energy and small size of the autonomous nodes, low computational power and frequent changes in topology [9]. Localization is needed in these networks to report the origin of events and to collect information about network coverage and connectivity [10].

Localization techniques for Tactical MANET (TMANET) include use of global positioning techniques, node connectivity fusion, time of arrival (ToA), and Direction of Arrival (DoA), as well as some nodes with known location information [10], referred to as anchors here. However, it may be unrealistic to equip every node with GPS devices or maintain GPS communication at all times, e.g. due to natural and architectural obstacles (e.g. dense forest, high raise building) that may interrupt line-of-sight signal reception from GPS satellites. Furthermore, as the energy efficiency is a crucial issue in MANET, the battery life of each node will be hampered due to power consumption of GPS. Another concerning issue is cost factor. For a large number of nodes the cost will increase dramatically.

Current location discovery techniques can be categorized in two basic approaches: (1) distance (or angle) estimation and (2) distance (or angle) combining. Various methods for calculating distance between two nodes exist. An elaborate discussion of these methods can be found in [11].

- *Received Signal Strength Indicator (RSSI):* In this method the power of the radio signal is measured at the receiver end and the effective propagation loss is calculated based on known transmitted power. Finally to estimate the distance, previously calculated loss is translated using theoretical and empirical models. This technique has been used specially for RF signal. The main advantage of this technique is that there is no extra device required as all the sensor nodes are usually equipped with radio. Due to the multipath propagation of radio signal RSSI performance is not as good as compared to the distance measuring method.
- *Time based Methods (ToA,TDoA):* This method mainly deal with time to calculate distance by recording time of arrival (ToA) or time –difference-of-arrival (TDoA). If the signal propagation time is known then this propagation time can be directly converted to distance. Like RSSI, this method is not only limited to RF signal. As well as RF, it can also be applied to acoustic, infrared and ultrasound signals. It has some drawbacks also. In some condition it is hard to fulfill the line-of-sight condition and the accuracy of signal detection can be hampered in case of acoustic signal due to its multi-path propagation effects.

- *Angle-of-Arrival (AoA):* Instead of measuring the distance AoA based systems mainly calculate the angle of received signal and node position is estimated by using geometric relation. The accuracy is higher than RSSI based method but there is a trade-off between the level of accuracy and the cost. Usually the hardware required for AoA method is more costly.

The second technique of location discovery approaches is mainly several combinations of first techniques and they can be classified in three main methods.

- *Hyperbolic trilateration:* By calculating the intersection of three circles as shown in Figure 1-2 (a) node's position is determined by this basic method.
- *Maximum Likelihood (ML) estimation:* The differences between measured distances and the estimated distances from known nodes are minimized by ML estimation to estimate the location of a node (shown in Figure 1-2 (b)).
- *Triangulation:* This method deal with the direction rather than calculating the distance as in an AOA system to determine the node position by using trigonometry laws of sines and cosines as shown in Figure 1-2 (c).

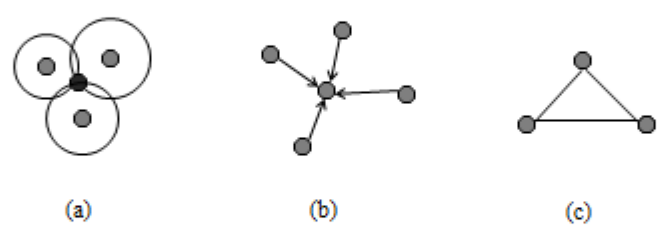


Figure 1-2: Location Discovery Approach.

The research community in the field of localization proposed many proposals and those can be classified in two broad categories based on the role and location of the situational awareness system:

- *Centralized Localization:* Rather than processing all ranging and connectivity information separately in each and every node of the network all the data is transferred to a central base station that is sufficiently powerful and capable to process all the data and send back the resulting location information to the respective nodes. There is a trade off in this centralized algorithm between the reduced complexity of computation in each node and the communication cost of migrating data to and from the base station.
- *Distributed localization:* In this algorithm all the computation process take places on the individual nodes and to locate their position in the network they communicate with each other. It can also be classified into several categories.

A classification of different approaches is shown in **Figure 1-3** [10].

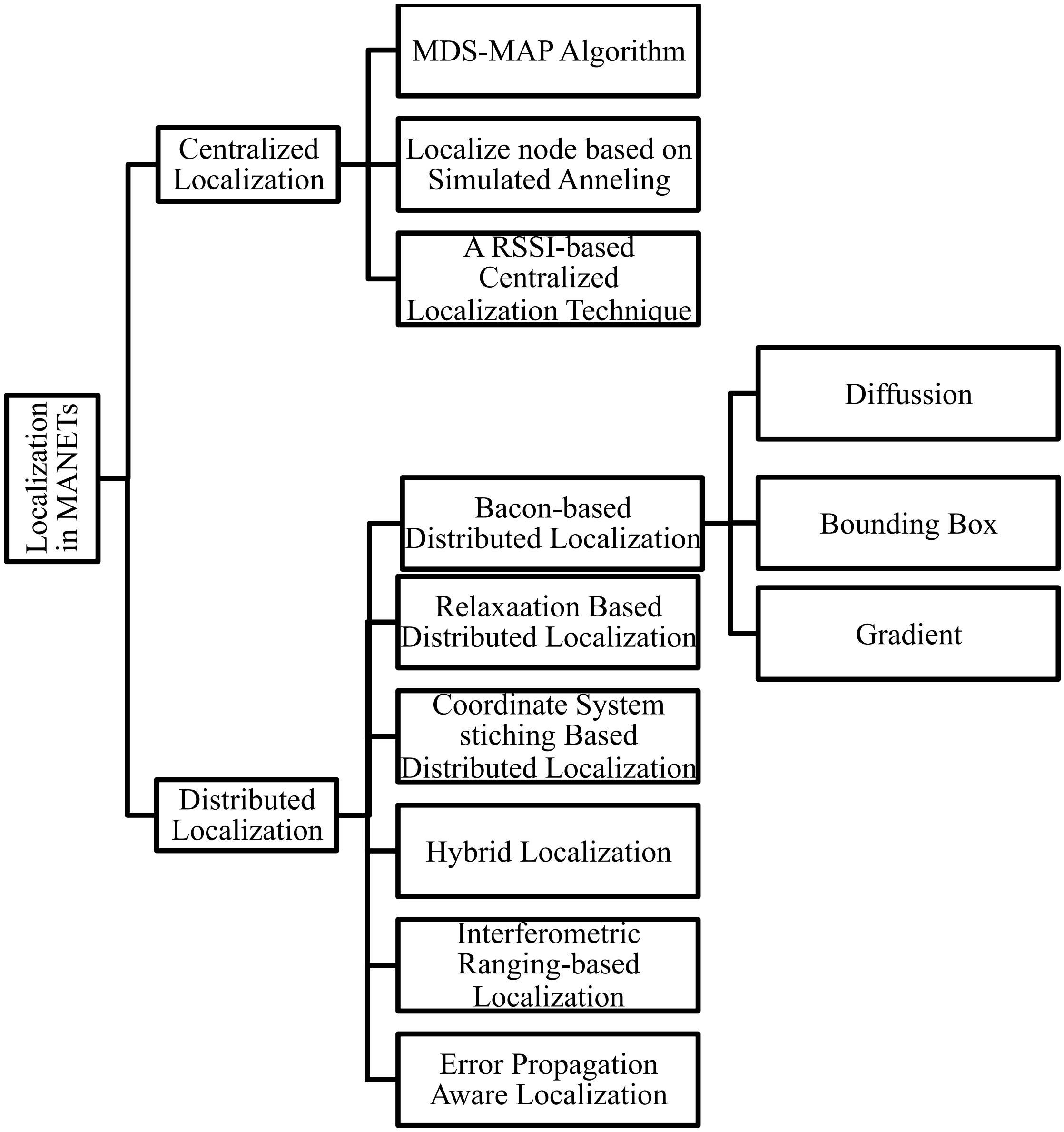


Figure 1-3: Classification of the different proposals for Localization.

1.2 **Related Works**

1.2.1 Visualization

Koyama et. al. [12] developed a visualization system called MANET-Viewer in order to observe ad-hoc behavior. The system follows a centralized approach by which there is a dedicated node (Personal Computer with wireless cards) for data collection. Using the flooding model used by a data collection application on the dedicated node performs the process of collecting data. The authors' claim that the performance of this tool depends on the nodes gathered information that varies depending on the stage of the network. Experimental results show that the node' information collection time increases as a number of hops increases. However, no results were presented regarding a number of nodes in the network. Also, this tool does not provide any form of location information except that a GPS coordinate can be entered manually.

Visualization of MANET simulations based on NS or OPNET has also been studied in literature. Typically NS-2 executes a network scenario and produces trace files containing time stamps of network events. Nam [13] and iNSpect [14] are network visualizers that playback NS-2 trace files. Nam was built for wired network visualization rather than the wireless network, and has some limitations in drawing wireless traffic. The iNSpect program was built to visualize wireless network events such as packet hops, wireless node links, and packet delivery success. iNSpect can only visualize wireless traffic. Belue et. al. [15] designed and developed an enhanced framework for visualizing military and non-military wired and wireless networks. The main idea of this framework is to animate events contained in a simulation tracefile. The acceptable trace files are NS-2 and OPNET. GTNets [16] is another network simulator that combines a full-featured network simulation environment with graphical viewing of the simulation. It shows the network topology along with network traffic. However, the visualizer component of GTNets is tightly coupled to the simulator and cannot be separated nor it can work with other simulators' outputs.

mLab is a MANET test bed from the National Institute of Standards and Technology (NIST) that addresses the balance between desktop simulations and outdoor field tests [17]. The motivation of the mLab came from the fact that simulation results may not accurately reflect in a real- world scenario; hence, a lab environment is necessary. mLab allows users to develop and test their MANET protocols and applications in a laboratory environment and takes advantage of simulated systems to bring the setup environment closer to actual deployment. It requires a set of PCs or laptops, each equipped with an Ethernet adapter and a wireless adapter. [18] is an extension of mLab to include a collection of host-based kernel network traffic monitoring modules.

The impact of a partial view of the network in data delivery in an OLSR network has been studied in [19]. This work was primarily focused on increasing data delivery rate by tuning various OLSR parameters and assessed the increase in control overhead.

In this thesis we present the framework for mobile ad-hoc network status and topology visualization based only on standard OLSR protocol databases, and without any modification to the standard OLSR operation [20]. In contrast to the work in [12], we develop an approach which can be used either in centralized command and control environment based on node queries, or in a non-intrusive manner based on the routing and topology information available at each node databases to build a map of the network, and use simulations to study the performance of centralized versus nodal visualization. In particular, we present results to show how quickly and with what degree of accuracy a nodal view of the network can approach the centralized full view.

1.2.2 Localization

Many current positioning techniques perform localization based on an existing infrastructure. Niculescu and Nath [21] and Nagpal et al. [22] proposed localization technique, which is mainly GPS, based triangulation. Both algorithms require landmark or beacon nodes. The rest of the nodes within the network calculate their hop distances to at least three beacon nodes to estimate their position. In [11] the authors proposed another distributed location discovery approach called AHLoS (Ad-Hoc Localization System) for WSN. In this approach only a few nodes are required to have exact location information and rest of the nodes will be able to dynamically discover their own location using proposed algorithm. For calculating the distance they used RSSI (for RF signal) and TDoA (For ultra sound transmission).

Nirupama et al. [23] also propose a low cost outdoor localization technique for very small devices without the help of GPS. This paper uses an idealized radio model which is mainly an RF based localization system, based on RSSI and connectivity respectively. Due to low performance of signal based approach they propose a simple positioning method based on connectivity.

Priyantha et al. [24] suggest a new distributed localization method based on force-directed algorithm known as the AFL (Anchor-Free Localization) and exploited the natural bridge with graph layout algorithm. This approach is totally different from the techniques discussed so far. For better accuracy this approach has to assume some additional information as there is no assumption available about angular or distance information or any fixed node with known location as anchor node.

All the localization techniques mentioned above work well under several conditions such as the topology should be static and predictable but their performance will be questionable for any network with continuously changing and highly mobile topology like MANET. Capkun et al. [25] propose a localization approach for MANET known as SPA (Self - Positioning Algorithm). This algorithm provides localization information without GPS and the scenario of the network is mobile and infrastructure less. Their proposed distributed algorithm is capable of finding location information only using local resources. Time of Arrival (ToA) techniques is used to calculate the distance between two nodes to create a network coordinate system. Kim et al. [26] also proposed a local positioning system for mobile ad hoc network using same techniques of [23] with some addition and instead of being proactive they propose a reactive mobile ad hoc positioning algorithm to reduce the network overhead to find the position a specific or few nodes.

Another localization algorithm for MANET called Adron is proposed in [7] which is symmetric and free of existing algorithms shortcomings such as flooding of the network, anchor or beacon nodes.

Wang et al. [10] presents an optimal fusion of time-of-arrival (TOA) and direction-of-arrival (DOA) technique for localization in MANET without any GPS information. This technique is a multi-node 2-Dimensional technique capable of optimizing positioning error in MANET with two types of nodes, which can communicate with each other. One is base-nodes furnished with antenna arrays to calculate TOA and DOA to locate other nodes within their range and other type of nodes are known as target nodes with omni-directional antennas.

In [27] the authors propose a centralized anchor free localization technique where the nodes are static and capable of reporting their range information. To find the location this range information is considered in one case and in other case in addition to the range information angular information is available. For the localization total six forces-directed algorithms are implemented and tested. Fruchterman-Reingold Algorithm (FR), Kamada-Kawai Algorithm (KK) uses only graph adjacency information and Fruchterman-Reingold Range Algorithm (FRR), Kamada-Kawai Range Algorithm (KKR) uses both graph adjacency information and range information. All these classical Force-directed Algorithms used the information contained within the graph itself to calculate the layout of the graph. If the network size is small and nods are distributed within a simple region these classic algorithm works well. But these simple force-directed algorithms couldn't construct the vertex location if the graph is larger than fifty. [27] Shows that Multi-scale Kamada-Kawai Range Algorithm (MSKKR) utilizes both graph adjacency and range information is capable of overcoming this problem. This algorithm mainly depends on a filtration of the vertices, intelligent placement and multi-scale refinement.

1.3 **Motivation and Objective**

Now a days from the military and industrial point of view wireless mobile ad hoc networks has rising application due to the enormous development of communication technologies in the military and civilian field that can't rely on traditional infrastructure base architecture. Due to the nature of MANET, they have the capability to act as a concrete solution in a crisis context to establish communication, like after a natural disaster [28] [29]. To create a common operative picture for a command and control center it is crucial to have the topology and location information of each component or node within the network and the aim of this thesis is to come up with a high level design for connectivity visualization of OLSR based MANET topology based on the local database available in an OLSR node. The focus of this thesis is to develop a situational awareness application that would provide a network common operating picture using connectivity and adjacency information from MANET routing protocols such as OLSR. The proposed approach takes advantage of the existing topology information exchanged among wireless nodes within the network.

Multi-hop wireless ad-hoc network is the most attractive among the wireless communication technology. Being specified as a self-organizing and a self-configuring network without any existing infrastructure, each and every single node within a TMANET must be capable of working as a router to forward messages from a sender node to other nodes that are out of sender's transmission range. Therefore, in MANETs routing protocol has an important impact on efficient data transmissions throughout the network. As routing protocol is essential for an ad hoc network, assigning a routing protocol for a network with several limitations such as limited bandwidth, change in topology and link failure due to mobility of the nodes is not a piffling one. Due to the limited acceptance range of topology range and higher control traffic the RIP and OSPF are not adapted and AODV, DSR and TORA also have the expensive flooding mechanism in terms of overhead and causing delays in route calculation [30]. While being a proactive routing protocol OLSR minimize overhead by periodically exchanging the message to keep up to date topological information of the network. Also the control traffic overhead is significantly reduced by OLSR compared to the other reactive routing protocol by using Multipoint Relay (MPRs) nodes. The delivery rate is also higher for

OLSR offering lowest delay in transmission [31]. The detail about OLSR is described in Chapter three.

One feature of the proposed situational awareness system design is to provide coarse node localization in TMANET. We propose a localization algorithm based on the Force directed method that will allow us to calculate the location of each node within the network relaying on network topology information with partial GPS or landmark information.

1.4 **Methodology**

This work proposes two approaches in visualizing network topology. Also, to support this assumption, this work further assumes that a visualization tool and its related computational will be on one of the nodes in the network. This also means that this particular node, a candidate node, will not have any form of power and processing limitations. Figure 3 shows an OLSR Tactical MANET scenario by which the candidate node will be the node that performs visualization related tasks. The scenario shown in Figure 1-4 will be implemented or modeled in NS-3. In the simulation model, in order to properly reflect real-world situation, a mobility model will be applied to the scenario.

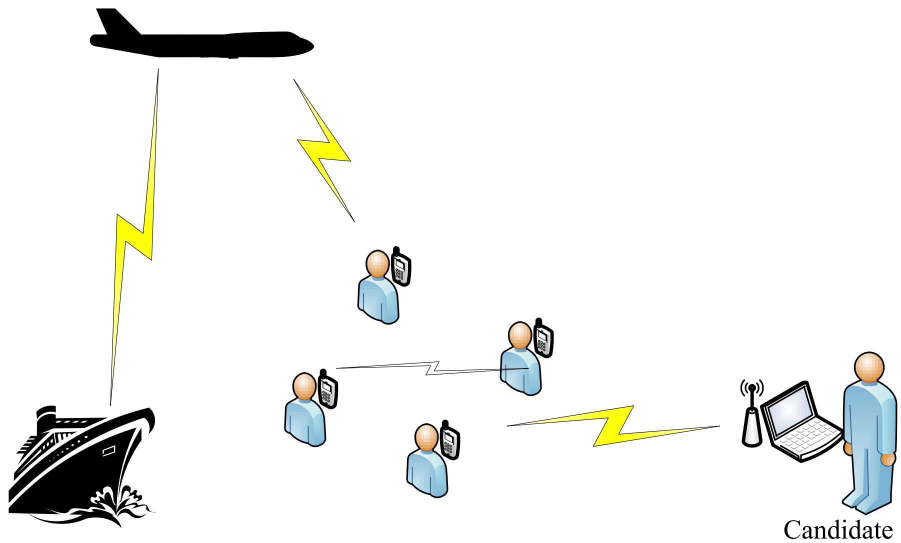


Figure 1-4 : The OLSR MANET scenario.

The two proposed approaches in visualizing network topology are centralized approach and nodal approach. In centralized approach situational awareness system would be running on command and control center or candidate node while in nodal approach each node is running its own situational awareness system. In centralized approach we assumed that the command and control center would receive or query the information from the network's node to build the topology map. The application process is required on the candidate node. The purposes of the application are to collect the node's topology information and location information and display the network topology graphically. There are two modules within this application: Topology and location file Generator and Visualizer. Figure 1-5 shows internal modules of the application running on the candidate node.

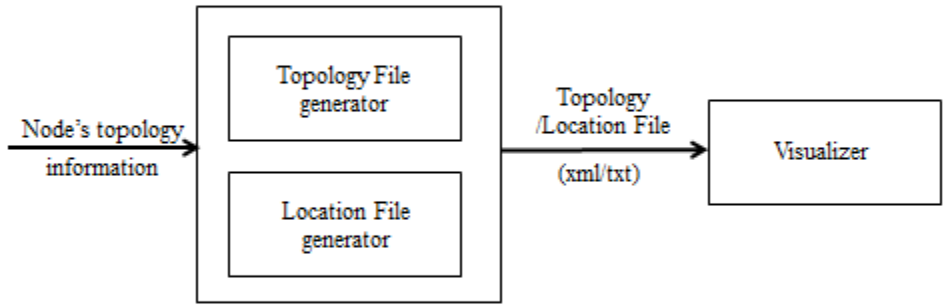


Figure 1-5: Visualization Process of the Candidate Node

The purpose of the topology file generator is to generate a topology file and update it in real-time. There are two possible approaches in implementing the topology file generator. The first approach is to allow the generator module to have access to the neighborhood database and topology database (routing layer) of the candidate node. The neighborhood database contains a set of tuples (N_neighbour_main_addr, N_2hop_addr) that contains the address of a node's 1 and 2 hops neighbor nodes. The topology database contains a set of tuple that stores the address of last hop node before the destination node and the address of the corresponding destination node in the network. The information read from these two databases will reveal the entire network topology. Note that each OLSR node also uses information from these databases in order to calculate its routing table. Another approach is to make the topology file generator module have access to all the incoming and outgoing topology messages. From these messages the topology generator module can create or construct topology information similar to the ones in the routing layer (neighborhood and topology databases).

In nodal approach we assume that the node themselves will build the topology map based on the information that they collect through their own routing protocol without any additional query or algorithm. Also in nodal view approach there is no additional overhead cause there is no need for querying others node's topology information. Note that OLSR does not provide information about links between non-MPR nodes or links that are not in its spanning tree. So we cannot build the whole network topology map using OLSR TC messages only. To resolve this issue we develop a custom message known as Network Information Message (NIM) to exchange 1-hop neighbor information in centralized approach. Depending on the scenario it can be query based or can be flooded throughout the network. Though it will increase the network overhead but will help us to discover whole network topology.

The topology information constructed in either approach will be parsed to and stored in a XML or text file. Note that this XML or text file will be updated in real-time in order to maintain the most of up to date network topology. The purpose of the location file generator is to calculate the location of each node using the proposed coarse localization algorithm based on node's topology information taking into account the GPS enabled node or the landmark if there is any and update it real time to feed the visualizer.

To examine the accuracy of network discovery OLSR information will be used. NS-3 simulation environment will allows us to extract the topology information of the whole network based on active links within the network at a particular time. Total number of connected links within a network can be calculated by taking into account the number of links with each neighbour node (including MPR nodes) and looking at the routing table of each node for links between each two hop neighbour and their neighbour, destination node and last hop before this destination.

1.5 **Thesis Outline**

The rest of the thesis is organized as follows. In Chapter 2, we provide a brief overview on OLSR explaining the core functionality and available database that is used throughout this thesis to discover network topology. The process we used to discover the network topology is presented in Chapter 3 followed by the proposed coarse localization algorithm in section 3.2. In Chapter 4, the analytical model for bandwidth analysis is presented followed by the performance evaluation of our thesis work in Chapter 5, where all the simulation and results are analysed and reported.

Finally the conclusion and future work is presented in Chapter 6.

Chapter Two: Optimized Link State Routing Protocol

In Mobile Ad hoc Network routing is a key issue. There are mainly two types of routing protocols: *proactive* and *reactive* routing protocol. In proactive protocol, the routing table is updated periodically to maintain route to all possible destination without the consideration that network traffic will go or not. But in reactive routing protocol the routes are discovered on demand and the routing table is updated only for the destination where the traffic is intended to go. The main advantage of proactive routing protocol is that it can distribute the information over the network very quickly. OLSR is a Link state routing algorithm that is optimized to reduce the communication overhead in a network. OLSR is proactive in nature. The purpose of this chapter is to review Optimized Link State Routing Protocol in MANET. More specifically focus on the topology discovery in a situational awareness context.

2.1 **Background - OLSR**

Optimized Link State Routing Protocol (OLSR) [20] is a routing protocol developed for mobile ad hoc networks (MANETs). It is a table driven and proactive protocol that minimizes the network overhead (the number of messages) in discovering nodes. OLSR achieves its optimization by using the concept of MPRs.

Multipoint relays are a group of selected nodes within the network, which are capable of forwarding any broadcast messages during the flooding process. But in case of classical flooding mechanism every single node is responsible for forwarding each message after receiving the first copy of the message that increases the network message overhead dramatically [20].

The MPR nodes are also responsible for generating link state information in OLSR and as a result the number of control messages flooded in the network is also reduced. In OLSR, partial link state information is distributed within the network as third optimization if the MPR nodes only report the links between itself and its selectors.

Conceptually OLSR topology discovery involves two phases: neighbourhood discovery and topology flooding. The concept of two-phase topology discovery is similar to OSPF [32] but the operations are not the same. In the first phase, using Hello messages discovers neighbour nodes. The exchange of Hello messages in OLSR also involves electing MPR nodes. MPR nodes are nodes that are responsible for broadcasting topology control (TC) messages (topology information), which would be flooded throughout the network, second phase. With this approach, only subset of the nodes in the network will be used to generate and broadcast topology messages (information) as compared to OSPF where there is no concept of MPR nodes and all nodes are responsible for forwarding (flooding) topology information. For calculating the routes this information is then used and optimal routes are provided in terms of the number of hops.

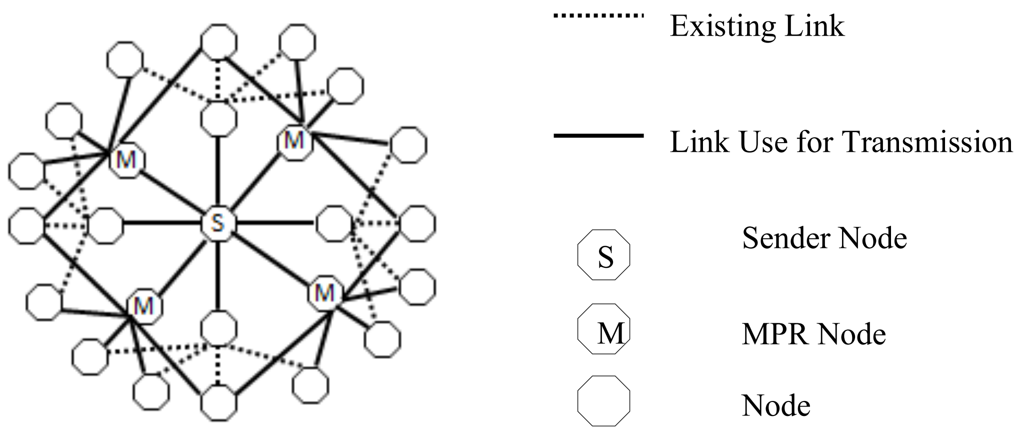


Figure 2-1: The OLSR flooding mechanism with MPR.

2.1.1 Advantages of OLSR

As it already mention being a proactive routing protocol OLSR minimizes overhead by periodically exchanging message to keep up-to-date topological information of the network. Also the control traffic overhead is significantly reduced by OLSR compared to the other reactive routing protocol by using one of his core functionality MPR. The delivery rate is also higher for OLSR offering lowest delay in transmission [31]. Aside of this there are also some advantages of OLSR mentioned in [33].

2.1.2 Addressing

For recognizing each and every single node within the network IP address is being used by the OLSR as an unparalleled identifier. An OLSR-enabled node can use more than one interface to communicate and that is the requirement to choose an IP address as its main address. IPv4 [34] and IPv6 [35] both are compatible with OLSR and the distinguishing factor among them in OLSR context is the size of the designated space in control message for the IP address.

2.1.3 Packet Format

As specified by the *Internet Assigned Numbers Authority* (IANA) to OLSR, it transmits all of its control traffic over UDP on port 698. While using IPv4, without specifying any broadcast address this traffic is broadcasted according to RFC 3626. OLSR packet is used for sending and receiving OLSR traffic within the network. The basic layout of OLSR packet is consisting of OLSR packet header and a message body as follows shown in Figure 2-2. More details can be found in [20].

32 bits

8 bits	8 bits	16 bits
Packet Size		Packet Sequence Number
Message Type	Vtime(Validity Time)	Message Size
Originator Address		
Time To Live	Hop Count	Message Sequence Number
MESSAGE		
Message Type	Vtime	Message Size
Originator Address		
Time To Live	Hop Count	Message Sequence Number
MESSAGE		

Figure 2-2: OLSR Packet Lay Out.

2.2 **OLSR Functionality**

According to RFC 3626, the OLSR functions are divided into two groups. One is the core functionality that is based on processing and generation of messages and the other is the auxiliary functionality based on situational demand. The features of these two functionalities are tabulated below:

Table 2-1: OLSR Functionality features.

Core Functionality Features	**Auxiliary Functionality Features**
Packet format and forwarding	Non-OLSR interfaces
Information and repositories	Link-Layer notifications
Main Address and multiple interfaces	Advanced link sensing
Hello messages	Redundant topology
Link sensing	Redundant MPR flooding
Neighbor detection	
Topology discovery	
Routing table computation	
Node configuration	

2.3 **Message Types**

There are mainly three types of messages defined by OLSR core functionality. However, to meet the demand of a designer the packet format of OLSR support for a wide variety of custom packets to be broadcasted and the default forwarding rule will be applied to transmit those unknown packet types.

2.3.1 MID Messages

If the OLSR enabled node has only single interface then the main address will be the single interface address. But in case of nodes with multiple interface addresses the Multiple Interface Declaration (MID) messages are exchanged to uniquely identify the node on the network by announcing one interface as "primary interface".

32 bits

8 bits	8 bits	16 bits
Packet Size		Packet Sequence Number
Message Type	Vtime	Message Size
Originator Address		
Time To Live	Hop Count	Message Sequence Number
OLSR Interface Address		
OLSR Interface Address		

Figure 2-3: MID Message Format.

2.3.2 HELLO Messages

For discovering topology the importance of exchanging HELLO messages is crucial in OLSR. This message is exchanged periodically only between nodes who are neighbors to each other. It is also used for link sensing and MPR selection.

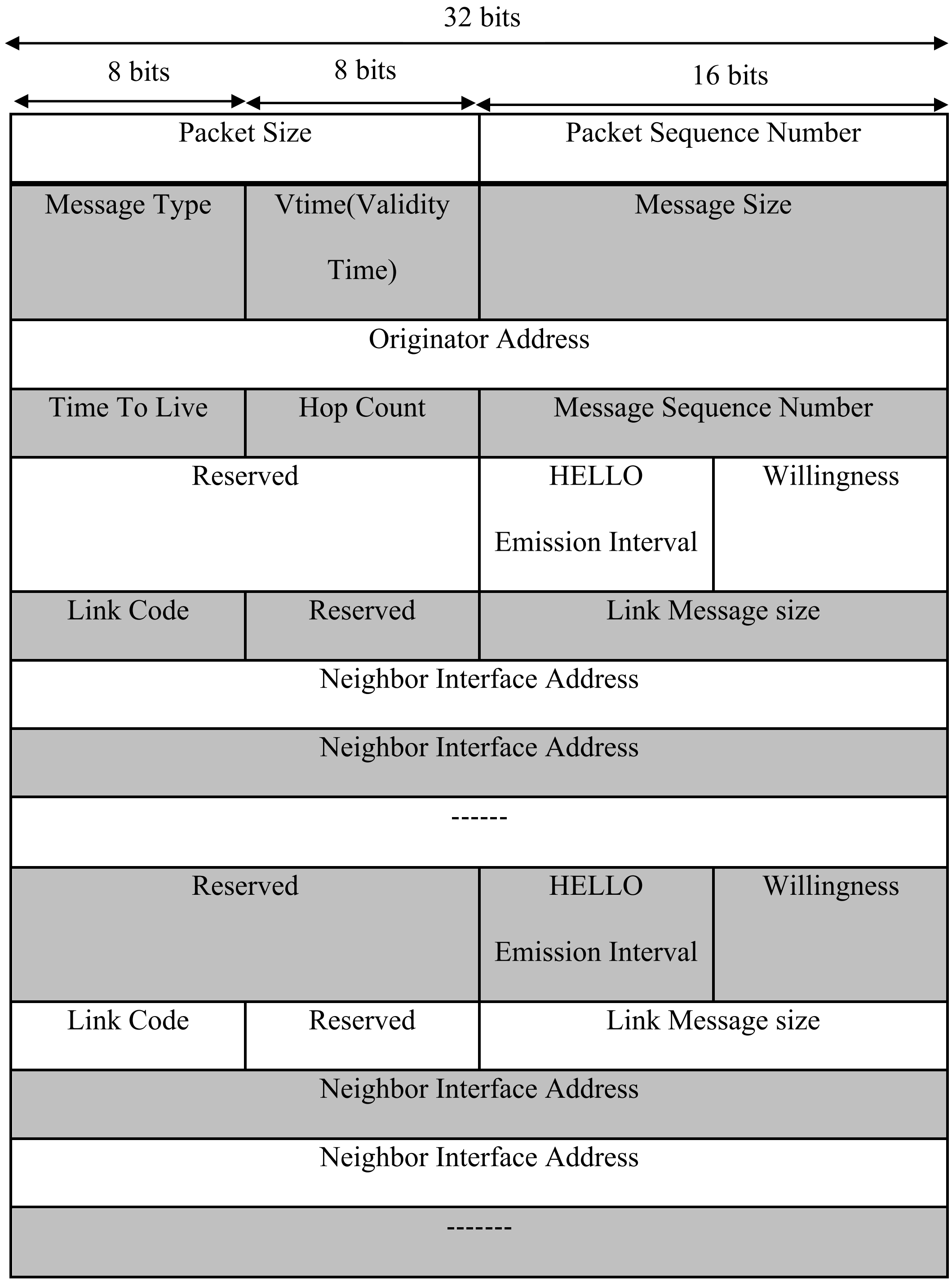


Figure 2-4 : HELLO Messages Format.

2.3.3 TC Messages

This Topology Control messages are responsible for building routing tables by flooding and spreading information regarding network topology by the MPRs only.

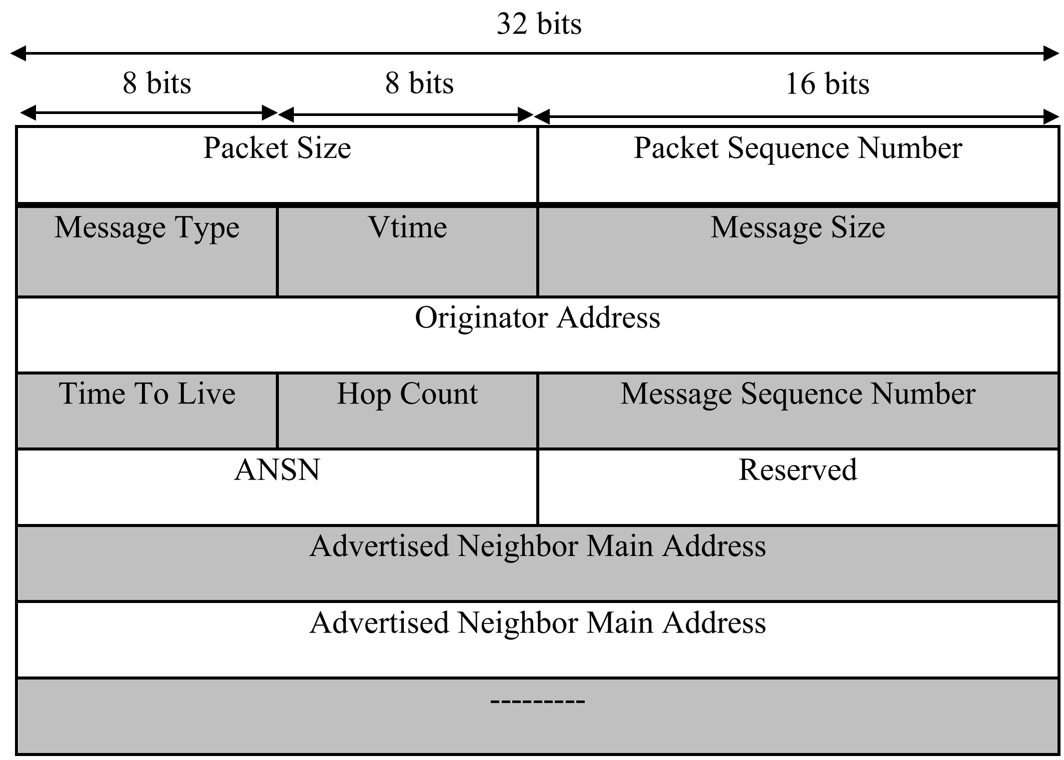


Figure 2-5 : TC Messages Format.

2.4 **Neighbor Discovery using OLSR**

Each node in the network with which it has a direct and bi-bidirectional link but there may be some uni-directional link due to the uncertainties over radio propagation must detect the neighbouring node. For identifying the neighbour node each node of a network broadcast *HELLO* messages periodically. During the broadcasting a list of all detected neighbour nodes and their link status is attached with the message. All one-hop neighbours of the sender node receive this *HELLO* message that is broadcasted once per refreshing period "*HELLO_interval*" but only forwarded by Multipoint Relays. This process discovers all one hop and two hop neighbours of a node. But there is an associated holding time for this neighbourhood and two-hop neighbourhood information after which it will be invalid [36].

The multipoint relay set is selected by an MPR selector node in a network among its one-hop neighbours in such way that all two-hop neighbours are covered by this MPRs. Details are described in [20]. Whenever a change is detected among one-hop or two-hop neighbourhood, the MPR set is computed.

2.5 **MPR Selection**

Each node calculates its own set of MPRs so that it can reach all its symmetric 2 hop neighbours via one of its MPRs. MPR calculation is based on willingness announced by neighbours using Hello messages. Willingness is one of the fields in a Hello message, which specifies the willingness of a node to carry and forward traffic on behalf of other nodes. According to the standard OLSR [20], willingness may be set to an integer value between 0 and 7. The willingness value of WILL_NEVER (an integer value of 0) means a node does not wish to carry traffic for other nodes and it will not be included in the MPR set. The willingness value of WILL_ALWAYS (integer value of 7) means a node is willing to or has resources to forward traffic for other nodes. Therefore, for a given node all the neighbour nodes with willingness equal to WILL_ALWAYS will always be included in the set of MPRs [20]. This is the beginning of the process and the node keeps adding its neighbours to the MPR set based on the reachability to the 2-hop nodes which is based on the link set, the neighbour set, and 2-hop neighbour set until all the 2-hop neighbours are covered by at least the one of the MPRs. The detail of the algorithm is described in section 8.3 of the RFC3626 [20].

It is worth mentioning that the willingness value of a node can be changed dynamically as the node's conditions change. A good example given in [20] is that as the battery of a node is drained the willingness value would be reduced. Also, by default as recommended by the standard [20], a typical OLSR node should be set to WILL_DEFAULT (an integer value of 3).

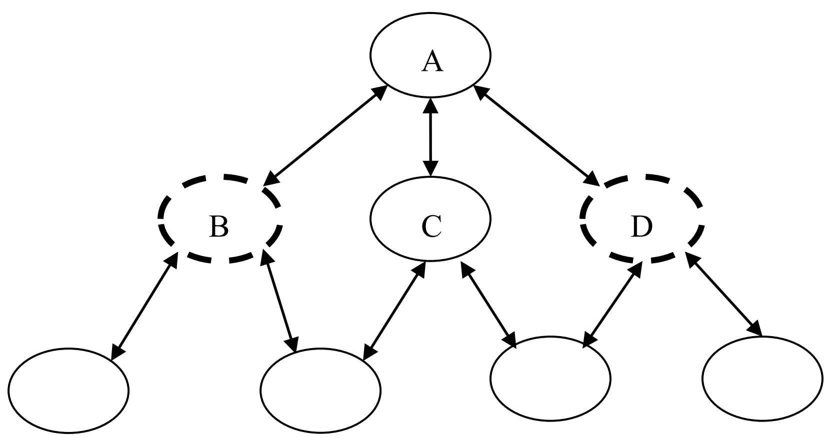


Figure 2-6: Selection of the MPR nodes B and D by the node A.

2.6 **Topology Discovery**

The operations of OLSR's topology discovery start with discovering neighbour nodes by exchanging *HELLO* messages. These *HELLO* messages are sent within 1 hop. Information embedded in these messages provides information to the receiving node of its neighbourhood within 2 hops. As part of exchanging *HELLO* messages, a node selects a set of its symmetric 1-hop neighbour nodes as MPRs. The selection criterion is that the symmetric 1-hop nodes are those that can reach the node's 2-hop neighbour nodes. In the perspective of an MPR node, those nodes that select this node as an MPR are called MPR Selectors. MPR nodes do keep records of their MPR selectors. The results of exchanging Hello message are records of nodes' 2-hop neighbourhood information, the next hop nodes to get to the 2-hop nodes (these are in fact MPR nodes) and MPR selectors. One may view that, for a given node, exchanging *HELLO* messages of OLSR reveals network topology of the area the node belongs within 2-hop radius.

The next operation in discovering topology is flooding *TC* messages. As mentioned previously, only MPR nodes are allowed to generate and forward *TC* messages. The information embedded in *TC* messages that each MPR node generates is at least the existing links between itself and its MPR selectors. The non-MPR nodes do receive *TC* messages, from their MPRs, and process them. However, non-MPR nodes do not forward (or flood) received *TC* messages.

The nature of OLSR reduces the number of messages exchanged in topology discovery. However, according to the standard OLSR [20], OLSR does not use any form of reliable mechanism in guaranteeing the reception of the topology information (messages) in contrast to OSPF that requires nodes to acknowledge when they receive topology information. Unreliable message transport in conjunction with nodes' movement and the random nature of the wireless signal may cause inaccuracy of topology information. In other words, topology information in a node may not reflect the current actual network topology accurately.

Standard OLSR [20] does give some flexibility in order to make OLSR more redundant by providing two tuneable parameters. These two parameters are TC_REDUNDANCY and MPR_COVERAGE. Typically, the TC_REDUNDANCY value is equal to zero which means an MPR node will advertise only links to its MPR selectors. An MPR node can also be tuned to advertise links to the combination of its MPR set and its MPR selector set, TC_REDUNDANCY = 1, or links to a full neighbour link set, TC_REDUNDANCY = 2.

Another tuneable parameter is MPR_COVERAGE. Typically, all 2-hop nodes must be reachable through at least one MPR node, MPR_COVERAGE = 1. Standard OLSR allows 2-hop nodes to be reachable through at least m MPR node, MPR_COVERAGE = m.

Both of these tuneable parameters provide redundancy to topology discovery in the OLSR network. In other words, there will be more topology messages exchanged in the network. [19] [37] [38] Have reported some simulations results that tuning these parameters (increase redundancy) does help increase accuracy of topology information and achieve data delivery.

2.7 **Information Repositories**

Being a table driven routing protocol, OLSR basically keeps all information regarding the current state of the network by maintaining a variety of databases of information. Each time after processing received control messages these databases is updated and as well as relies on this tables to generate this messages. Various information repositories used by OLSR are mentioned below:

- *Multiple Interface Association Information Base*: If there is any node within the network that has more than one interface then this information is stored in this database along with the interface address of such node.

- *Link sensing*: *Local link Information Base*
    - *Link Set*: Information regarding the state of links to neighbours is maintained in these repositories and calculated.
- *Neighbour Detection: Neighbourhood Information Base*
    - *Neighbour Set*: All the one-hop neighbour information (both symmetric and asymmetric) is kept in this database and based on the information in the link set it is updated.
    - *Two*-hop *Neighbour Set*: All 2-hop neighbours that can be reached via 1-hop neighbour are listed here.
    - *MPR Set*: All MPRs neighbours selected by one node are tabulated here in this database.
    - *MPR Selector set*: This is the repository that contains the neighbour information that selects a node as MPR.
- *Topology Information Base*: This database contains information regarding the topology of the network and acquired from TC messages to maintain the routing table.
- *Duplicate Set*: Information regarding recently processed and forwarded messages is stored here.

Chapter Three: Topology Discovery and Approximate Localization with Respect to Situational Awareness

The previous chapter provided a review of OLSR protocol and in this chapter, a brief description of topology discovery approach using OLSR database and information for visualizing the network topology in order to provide a better situational awareness system is given. Also our proposed coarse localization algorithm based on a Force-directed method that will allow us to estimate the approximate location of each node based on network topology information from a local OLSR database is discussed as well.

3.1 **Topology Discovery**

Network topology discovery is the main step in connectivity visualization in situational awareness context. A network is modeled as a graph of nodes with connections established between the pair of nodes. It can be assumed that in a wireless MANET, all authorized nodes will establish connections to all other nodes within the range of their transceivers but some networks may have security policies that limit such access based on the trust level of the nodes [39]. Furthermore, the transmitting range of a node may not be necessarily the same as its receiving range. Such issues are typically dealt with at the layers below the network layer and as a result their impact is already included in the routing tables, i.e. no direct route will be considered between two nodes that are not authorized to connect directly.

Most current mobile ad-hoc network visualization tools fall into one of the two categories: position plotting tools (often based on GPS data), or graph visualization tools that display the status of connectivity between nodes based on routing table data [39]. Our focus in this research is on the connectivity visualization based on the topology and routing data available from the routing protocol at each node.

In nodal view approach the situational awareness system make use of OLSR topology information that OLSR had already collected. But in centralized approach situational awareness system run on a central node or a command and control center that query the rest of the nodes within the networks for their neighbour information.

3.1.1 OLSR information to discover topology

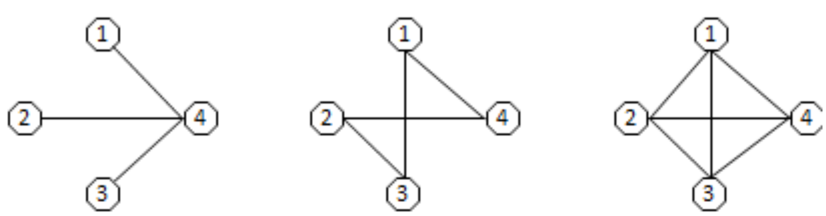
Conceptually the OLSR topology discovery involves two phases: neighbourhood discovery and topology flooding. The concept of two-phase topology discovery is similar to OSPF [32] but the operations are not the same. In the first phase, using Hello messages discovers neighbour nodes. The exchange of Hello messages in OLSR also involves electing MPR nodes. MPR nodes are nodes that responsible for broadcasting topology control (TC) messages (topology information) which would be flooded throughout the network, second phase. With this approach, only subset of the nodes in the network will be used to generate and broadcast topology messages (information) as compared to OSPF where there is no concept of MPR nodes and all nodes are responsible for forwarding (flooding) topology information.

3.1.2 Steps for Topology Discovery in OLSR

The results of exchanging Hello message are records of nodes 2-hop neighbourhood information, the MPR nodes and the MPR selectors. One may view that, for a given node, exchanging Hello messages of OLSR reveals network topology of the area the node belongs within 2-hop radius. The next step in discovering topology is flooding TC messages. Only MPR nodes are allowed to generate and forward TC messages. The information embedded in TC messages generated by an MPR includes at least the existing links between itself and its MPR selectors. The non-MPR nodes do receive TC messages from their MPRs and process them. However, non-MPR nodes do not forward the received TC messages. This feature of OLSR reduces the number of messages exchanged in topology discovery.

3.1.3 Adjacency Matrices

An adjacency matrix is a way of representing the connectivity between different vertices of a graph or which vertices (or node) of a graph are adjacent to which other vertices. The topology of a TMANET network can be represented by a graph $G = (V, E)$ where$V$, a set of vertices represents the nodes within the network and E is a set of unordered pairs $(u, v), u, v, \in V$ known as edges or links that represents the connection between the vertices or nodes [41].



$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \qquad \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix} \qquad \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

(a) (b) (c)

Figure 3-1: Example of Adjacency Matrix.

The adjacency matrix $A = [N_{ij}]$ for the N node network, each element in the adjacency matrix specifies the condition of a link in the network; i.e. if $L_{ij} = 1$, then an active link between nodes N_i and N_j exist and if $L_{ij} = 0$, there is no link between the two nodes (i.e. they are out of each other's range). If all links are bidirectional, the adjacency matrix in this case will be symmetric. Also the diagonal elements of the adjacency matrix is always equal to zero, $N_{ii} = 0$ for all i, and for $i \neq j$,

$$N_{ij} = \begin{cases} 1, & link\ i \rightarrow j \text{ exists} \\ 0, & link\ i \rightarrow j \text{ does not exist} \end{cases}$$

In centralized approach we assume that a node will be connected to other nodes if they are within each other communication range and this information might be collected by OLSR or may be using layer-2 adjacency information as soon as they establish ad-hoc connection. By adjacency we mean the nodes that are connected over communication link. The adjacency matrix is updated periodically as OLSR updates its tables. At each instant once the adjacency matrices were built for each nodal_view (distributed view), and combined to create the full_view (centralized view). In a mobile network, the number of link is dynamic; i.e. it changes as nodes move around and establish or lose connections.

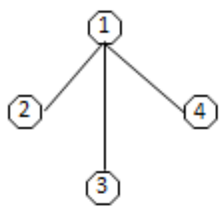
Suppose at a given time *t*, network nodes are positioned at locations$\mathrm{L} = \{\mathrm{L}_{1,}\ \mathrm{L}_{2,}\ \mathrm{L}_{3,} \dots \mathrm{L}_{\mathrm{N}}\}$. Let the x, y coordinates of node i position is denoted as $\mathrm{L}_{\mathrm{i}}^{\mathrm{x}}$, $\mathrm{L}_{\mathrm{i}}^{\mathrm{y}}$ respectively. Also supposes R= {R_1, R_2… R_N} represents the transmission range of each node. For simplicity we assume that each node will establish ad-hoc links with all nodes within its communication range and no link with any node outside of it.

OLSR adjacencies are established over links to neighbours. An N×N adjacency matrix $A = \{a_{ij}\}$ can be defined in the following manner:

$$a_{ij} = \begin{cases} 0 & i = j \\ 1 & i \neq j \text{ and } d_{ij} \leq R_i \\ 0 & i \neq j \text{ and } d_{ij} > R_i \end{cases} \qquad (1)$$

$$where \quad d_{ij} = \sqrt{(L_x^i - L_x^i)^2 + (L_y^i - L_y^i)^2}$$

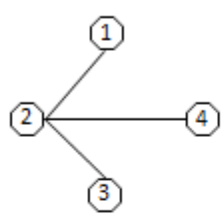
The adjacency matrix given for third network topology in Figure 3-1(c) is for a full or centralized view at any instance of time. Now, at the same time if we want to create an adjacency matrix for each nodal view than the topology and matrixes will be as follows:



$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

(a) Network Topology (b) Adjacency Matrix

Figure 3-2: Network Topology and Adjacency Matrix for nodal_view (Node 1)



$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

(a) Network Topology (b) Adjacency Matrix

Figure 3-3: Network Topology and Adjacency Matrix for nodal_view (Node 2).

3.1.4 OLSR Databases to Create Adjacency Matrices

As we discussed in the previous chapter, being the table driven routing protocol, OLSR basically keep all information regarding the current state of the network by maintaining a table. To generate the adjacency matrices for full_view or nodal_view, we only need access to those information repositories that contain information regarding active links between two nodes and those are:

- Neighbour Set.
- 2-hop Neighbour Set
- MPR Set
- MPR Selector Set
- Topology Database.

In our proposed nodal view approach, the generator module has access to the neighbourhood database and topology database of the node. The neighbourhood database contains a set of tuples (neighbour_main_addr, 2_hop_addr, and time stamp) that contains the address of a node's first and second-hop neighbour nodes. The topology database contains the address of last hop node before the destination node and the address of destination nodes in the network. The information read from these two databases reveals the nodal view of the network topology. If the eccentricity of a node is less than four hops, the OLSR neighbourhood and topology databases at that node provide sufficient information to identify all network nodes. If a node is four hops or more from a destination, than the 1st-hop, 2nd-hop and last hop tuple will not be sufficient for identifying all intermediate nodes to that destination. If a centralized approach is used, i.e. the topology generator has access to databases on all nodes, and then a complete view of all nodes can be constructed using the 1st- and 2nd-hop neighbour databases only.

3.2 **Localization**

To create a common operative picture for a command and control center it is crucial to have the location information of each components or nodes within the network. All the methods discussed in section 1.2.2 require nodes that are able to measure parameters such as ToA, DoA and RSSI, and share such information with the situational awareness system at the command and control centre for visualizing the network topology. In addition to an increase in cost, collection of all this information from the network also requires additional overhead bandwidth and consumes significant power. In this thesis we propose an alternative approach to approximate the location of nodes using the layer-3 connectivity data provided by link-state routing protocols such as OLSR, assuming that the range of the nodes are approximately known in advance. While as discussed later, the accuracy of this method will depend on the range of the nodes and mesh-ness i.e. connectivity information of the network, we show that a coarse localization using a combination of layer-3 connectivity with partial static or landmark information could be feasible. This approach would require much less control bandwidth, i.e. exchange of adjacencies instead of numerical values of DoA and ToA; and therefore can be used as a first stage localization which, if necessary, can be refined for selected nodes through distance or angle-based methods.

3.2.1 Problem Definition

In order to define the localization problem in OLSR-based Tactical MANET, we can consider a scenario of an area (e.g. battlefield) where N nodes (communication devices) are deployed. Each node establishes ad-hoc links with all network nodes within its range.

We assume that the total area is substantially larger than the range of a single node; therefore, multi-hop communication is needed. All nodes are running OLSR routing protocol to create this multi-hop ad-hoc network, and each maintains its local copy of its 1st-hop and 2nd-hop OLSR neighbours [8]. The nodes are mobile; therefore the network topology changes constantly. The overall ad-hoc network is represented by a graph $G = (V, E)$ where $V = \{v_i\}$ a set of N vertices representing the nodes within the network and E is a set of unordered pairs $(u, v), u, v, \in V$ known as edges or links that represents the connection between the vertices or nodes [42].

3.2.2 Force-Directed Method for Localization

As our localization problem can be treated as graph layout problem. The Force–directed method is used to resolve the problem as it is the most popular way of drawing graphs. The graph is considered as a physical model where vertices act as particles with active force in between [41]. Depending on the layout of the graph the forces are calculated.

There are two parts for this force-directed model: the model that describes the procedure for transforming a graph into a system of physical force defining the type of acting force in it and the second part is the algorithm seeking an equilibrium state for the model by configuring the system in such a way that the total energy of the system or the acting force on vertices within the graph will be as small as possible [7]. The initial starting of the configuration process by the algorithm can be totally random or some preliminary algorithm can be used to create initial configuration or the vertices can be placed manually to find the final equilibrium condition of the physical system [43] [44]. Tutte [45] first introduced this Force-Directed Method to draw graph. The method of this model is pretty straightforward: if there are two adjacent vertices then a force is imputed to each edge, which is proportional to the distance of them. So, if there are two vertices u, v and L_u, L_v is the position vector respectively then the total force on each of them will be

$$F\ (v) = \sum_{(u,v)\in E}(L_u - L_v) \quad (2)$$

For such a system, the equilibrium condition can be achieved when the position of vertices is calculated by satisfying the condition

$$F\ (v) = \sum_{(u,v)\in E}(L_u - L_v) = 0 \quad (3)$$

For 2 D scenario, this can be expressed as

$$F_x\ (v) = \sum_{(u,v)\in E}(x_u - x_v) = 0 \quad (4)$$

$$F_y\ (v) = \sum_{(u,v)\in E}(y_u - y_v) = 0 \quad (5)$$

Though [45] categorized force-directed as a method of graph drawing but [46] first published the practical force-directed model with the idea of spring and repulsive particles those are mutual to each other. For adjacent edges a logarithmic spring is placed among them and similarly for non-adjacent vertices a repulsive force is defined. As there are many springs and repulsive forces they need to be balanced and to adjust this balance [46] also introduce two constants k_{uv} and c_{uv} . Total force, F can be computed using following equation:

$$\boldsymbol{F\ (v) = \sum_{(u,v)\in E} k_{uv}\ \log\left(\frac{|L_u - L_v|}{l_{uv}}\right)\frac{L_u - L_v}{|L_u - L_v|} - \sum_{(u,v)\notin E}\frac{c_{uv}}{|L_u - L_v|^2}\frac{L_u - L_v}{|L_u - L_v|}} \quad (6)$$

Where l_{uv} denotes the preferred distance between u and v.

There are some notable force-directed methods have been published and all those are inspired by [46]. Two of them are closely related to our work for establishing an algorithm to resolve the localization problem in OLSR based Tactical MANET is presented below:

- ***Fruchterman – Reigngold (FR) Force- Directed method :***

In spite of similarities in logic with [46] the FR method [47] can be easily distinguished due to its different physical metaphor. For adjacent vertices an attractive force function and similarly for non-adjacent vertices a repulsive force function is clearly defined by FR method. The Euclidean distance d within two vertices are measured to calculate the magnitude of forces and are expressed as

$$f_a(d) = \frac{d^2}{l}, \quad f_r(d) = -\frac{l^2}{d} \quad (7)$$

Where f_a and f_r is the notation of attractive and repulsive force respectively. In case of adjacent vertices

$$f_a(d) + f_r(d) = 0 \;=>\; \frac{d^2}{l} - \frac{l^2}{d} = 0 \;=>\; \frac{d^2}{l} = \frac{l^2}{d}$$

$$=>\; d^3 = l^3 \;=>\; d = l, \quad d, l \in R \quad (8)$$

And so l is the ideal distance among them. Finally by using Eq. (7) the total force acting on any vertex □ can be calculated as

$$F\,(v) = \sum_{(u,v)\in E} f_a\,(|L_u - L_v\,|)\frac{L_u - L_v}{|L_u - L_v\,|} + \sum_{u\in V,\ u\neq v} f_r\,(|L_u - L_v\,|)\frac{L_u - L_v}{|L_u - L_v\,|}$$

$$= \sum_{(\boldsymbol{u},\boldsymbol{v})\in \boldsymbol{E}} \frac{|\boldsymbol{L_u} - \boldsymbol{L_v}\,|^2}{\boldsymbol{l}}\frac{\boldsymbol{L_u} - \boldsymbol{L_v}}{|\boldsymbol{L_u} - \boldsymbol{L_v}\,|} - \sum_{\boldsymbol{u}\in \boldsymbol{V},\ \boldsymbol{u}\neq \boldsymbol{v}} \frac{\boldsymbol{l}^2}{|\boldsymbol{L_u} - \boldsymbol{L_v}\,|}\frac{\boldsymbol{L_u} - \boldsymbol{P_v}}{|\boldsymbol{L_u} - \boldsymbol{L_v}\,|} \qquad (9)$$

Placing the vertices within the drawing area randomly using FR algorithm to find the final equilibrium condition of the physical system starts the initial configuration process. During iteration process, the amount of net force vector associated with each vertex is moved per iteration. If there is any certain threshold for the maximum amount of net force acting on a single vertex then the iteration process continues until it reached below this threshold amount. This process can also be stopped after having a maximum number of iterations.

- ***Kamada-Kawai (KK) Force-Directed method :***

The graph-theoretic distances of vertices can be considered as Euclidean distances for solving the graph layout problem. A force-directed method is proposed by [48] which is a spring based pure energy model. According to this model each pair of adjacent and non-adjacent vertices are affiliated by a Hooke’s law spring and the length of each spring is proportional to the graph-theoretic distance of the two associated vertices.

According to Hooke’s law the total energy of a single spring is

$$E = \frac{1}{2}k_s d^2 \qquad (10)$$

Thus, the energy in a KK system can be simply expressed as

$$E = \sum_{u,v\,\in V,\ u<v} \frac{1}{2}k_{uv}(|L_u - L_v\,| - l_{uv})^2 \qquad (11)$$

Wherek_{uv} and l_{uv} denotes the spring constant and preferred spring length between u and v repectively.

The proposed initial configuration process is started by placing the vertices evenly on a circular arc in random order for finding the final equilibrium condition of the physical system Newton -Raphson iteration method is used by the KK algorithm in order to reduce the amount of energy of the system as small as possible to achieve the equilibrium state. During each step of the iteration a new position for one vertex is determined at a time.

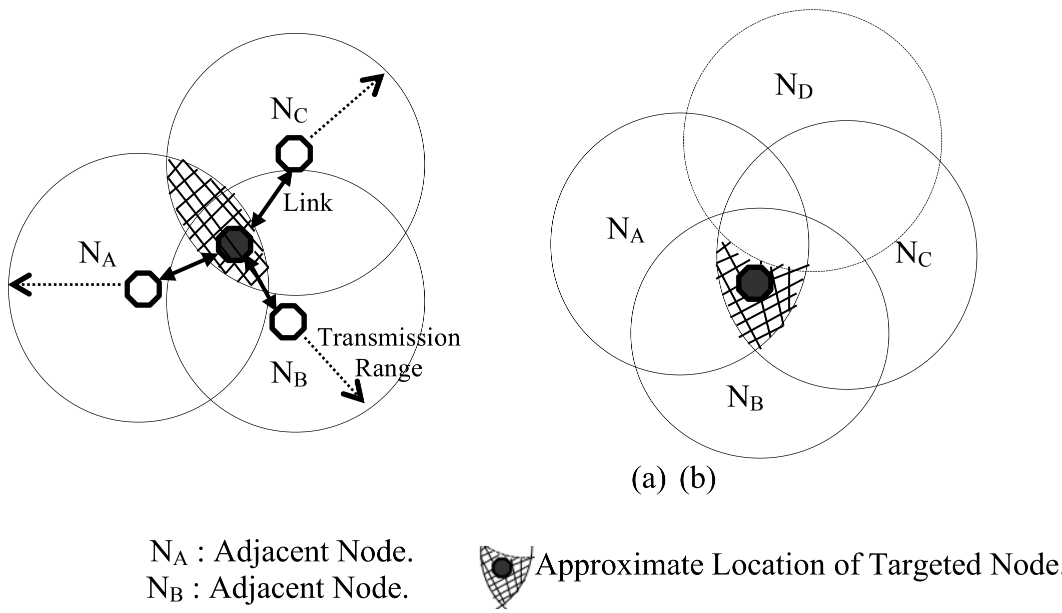
3.2.3 Coarse Localization

Our proposed method has the task to calculate the approximate physical location of the node. Our proposed technique is non-trivial for several reasons:

To calculate the approximate position,

a node should know

- The connectivity information.
- Must be within the transmission range of other nodes.



.

Figure 3-4: Coarse localization using Topology information.

Each node in the OLSR network is able to build the adjacency matrix of the entire network based on the OLSR link state advertisements it receives from other nodes of the network. Once the adjacencies are known, it may be possible to approximate the residing area of a node based on its transmission range and the nodes it connects too. Suppose S_j denotes the transmission area of node $v_i \in V$, and $\overline{S_j}$ denotes the area outside the transmission range of this node. Then as shown in Figure: 3-4, the approximate residing area of a node must include the overlap of the transmission areas of all its adjacent nodes, and excludes the transmission areas of all its non-adjacent nodes. In other words:

$$L_i \in (\bigcap_{a_{ij}=1} S_j) \bigcap (\bigcap_{\substack{a_{ij}=0 \\ i \neq j}} \overline{S}_j) \qquad (12)$$

Obviously, this approach could only provide useful information if a true multi-hop environment exists; i.e. presence of a sufficient number of adjacent and non-adjacent nodes. For each pair of nodes, we can only determine if they are within each other range (Ri+Rj). If the network is a full mesh, i.e. all nodes are within the transmission range of each other, the accuracy of this method would be limited to 2*R (the total diameter of the area of the transmission range). We will show later that the accuracy of this approach deteriorates when the network density increases. However in dispersed mobile networks in a sufficiently large area, an approximate residing area can be identified, which we refer to as coarse localization.

It may seem that in order to calculate the approximate residing area for node v_i based on Eq. (12), the exact location of all other nodes in the network must be known. However we assume that only a subset of network nodes may be GPS-Enabled (GE) or close to a landmark with known coordinates. The location of the rest of the nodes is unknown at the beginning of the algorithm. In the following we introduce the Force-Directed algorithm that allows us to calculate the approximate residing areas of all non-GE and non-landmark nodes based on the concept of Eq. (12).

To achieve our proposed goal a network visualizer tool was designed in an object oriented fashion along with multiple threads that is capable of calculating nodes coarse location using our proposed algorithm based on OLSR topology information and visualize the network topology along with the calculated position of the nodes.

For our system we use a variation of the FR method, which a link between adjacent nodes is modeled as a spring and each node is modeled as an electron. Therefore Hooke's law describes the attractive force while the repulsive force is determined based on Coulomb's law:

$$\mathbf{f}^{\mathbf{r}}_{\mathbf{x,y}} = \mathbf{k_e}\frac{\mathbf{Q_1 Q_2}}{\mathbf{d^2}} \quad (13)$$

$$\mathbf{f}^{\mathbf{a}}_{\mathbf{x,y}} = \frac{\mathbf{1}}{\mathbf{2}}\mathbf{k_s}\boldsymbol{d^2} \quad (14)$$

Coulomb's law states that the force between two particles is inversely proportional to the distance squared between them, while Hooke's Law states that the force between two particles is linearly proportional to the distance between the nodes. The net force for each axial dimension is the summation of all forces interacting with a targeted node. Before any reposition calculations are applied each node is refreshed with the net force for each dimension as shown in Eq. (15) the parameters in Eq. (13) and Eq. (14) are fine tuned to minimize convergence time and accuracy.

$$\boldsymbol{F_x} = \sum_{\forall\, i \neq x}^{N} \left(\boldsymbol{f^r_{x,i}} + \boldsymbol{f^a_{x,i}}\right) \quad (15)$$

Below in Figure: 3-5 represents one of our first threads developed to expand and contract our network mesh by using Force-Directed algorithms to achieve coarse localization of mobile nodes. Our first step of the algorithm resulted in simple tactics of attaching springs to each adjacent node and treating each non-adjacent as an electron for ∀ nodes.

- At the start of the configuration process, the algorithm checks for any GPS-Enabled nodes to act as anchor nodes. The anchors' positions are known.
- Then the non GPS-enabled nodes are randomly distributed.

- As we are relying on layer-3 connectivity information to calculate coarse location the adjacency information is also provided to our algorithm. Based on the adjacency the links between the nodes are established and nodes are connected to each other.
- The distances between the nodes are calculated. Before applying force to the nodes we need some other values for the co-efficient of Hook's and Coulomb's law. These co-efficient values are determined in trial and error basis to minimize the convergence time and remain same throughout the process and shown in Figure 3-5.
- After that depending on the connectivity matrix the force is applied. If there is any active link between two nodes and they are within transmission range of each other than attractive force is applied and the repulsive force is applied if they are outside each other transmission range.
- Depending on the Hook's law and Coulomb's law nodes will pull them closer together or push them further apart. This is repeated iteratively until the forces acting on the system come to an equilibrium state; i.e., the relative positions of the nodes do not change anymore in next iteration. The physical interpretation of this equilibrium state is that all the forces are in mechanical equilibrium.
- At this moment, the approximate locations of the nodes are calculated. As it is a mobile scenario we are dealing with snapshot of the network that is taken in an instant interval of time and this coarse localization approach is looped back to next snapshot of the network.

The pseudo code is as follows:

1. Spring_coefficient, K_s =0.05
2. Electrostatic_coefficient, K_e = 10000000000.0
3. Electron_charge, Q_n = 1.0
4. Changed position of the nodes due to axial forces = d_x and d_y
5. Setup initial node velocities d_x and d_y = 0
6. Setup initial NON-GPS node position randomly
7.   for each u in V
8.   Do for each v in V such that $u \neq v$
9.   Do for each z in E, $u \in z$ and $v \in z$
10.   Calculate Euclidean Distance, d.
11.   Compute $f^{a}_{x,y}$ using Eq. (14)
12. $d_x = d_x + f^a * (u_x - v_x)$/Distance
13. $d_y = d_y + f^a * (u_y - v_y)$/Distance
14.   For each z in E, $u \in z$ and $v \notin z$
15.   Calculate Euclidean distance, d.
16.   If $d < 3$ *communication range
17.   Compute $f^{r}_{x,y}$ using Eq. (13)
18. $d_x = d_x + 0.5 * f^r * (u_x - v_x)$/Distance
19. $d_y = d_y + 0.5 * f^r * (u_y - v_y)$/Distance
20.   for each u in V
21. $u_x = u_x + dx$
22. $u_y = u_y + dy$
23.   if $|d_x|, |d_y| < \varepsilon$
24.   Then end
25.   else go back to step 8.

Figure 3-5: Localization using Force-Directed Algorithms.

The adjacency matrix by itself cannot provide the absolute location information; it has to be supplemented with anchors. By anchors, we refer to a subset of nodes whose exact location information is provided outside of the FD algorithm; for instance nodes that are GPS-enabled or near a landmark. The existence of these nodes would allow the algorithm to pin down the location of other nodes. A trade off exists here between the number of anchors (that adds to the complexity of the system) and the accuracy of locations. These selected nodes only provided electrostatic forces for nodes whether or not they are attached to other nodes with springs. Throughout the process of minimization each node is associated with a GPS coordinate relative to a default position supplied during program initialization. As a result the physics engine relies on the GPS coordinate system for calculations similar to the KK algorithm as both rely on theoretical values for distances between nodes. The pseudo code is as follows

Unlike the (FR) [46] algorithm and (KK) [47] algorithm the field of view for each node is defined by a free-space model where each node is only aware of their adjacent neighbours and an adjustable view of the field to scan for nearby nodes.

For us it was important to keep in mind that these disconnected nodes in mobile mesh networks are very sporadic, chaotic, and random. Since one of the qualities of MANET networks is the inconsistency of nodes connectivity and response to position information, a MANET situational awareness system can use a simple caching service to remember disconnected nodes for a period of time. This service will store the last known position of a lost node and flags it as inactive until it hears back from it.

3.2.4 Comparison of Our proposed Method with FR and KK Method

In [27] the authors propose a centralized anchor free localization technique where the nodes are static and capable of reporting their range information. To find the location this range information was considered in one case and then other case was considered in which in addition to range, the angular information is also available. Various underlying region for the network was also taken into account (eg, Simple convex polygons, non-simple polygons, simple non-convex polygons).

In [27], for localization total six force-directed algorithms are implemented, tested and their performance are evaluated. FR [46] and KK [47] algorithms used only graph adjacency information and Fruchterman-Reingold Range Algorithm (FRR) [25], Kamada-Kawai Range Algorithm (KKR) [25] used both graph adjacency information and range information. All these classical Force-directed Algorithms used the information contained within the graph itself to calculate the layout of the graph. For adjacent vertices an attractive force function and similarly for non-adjacent vertices a repulsive force function was clearly delimited by FR algorithm. Finally a minimum of this function was determined to compute a layout for the graph. Nodes theoretical distance within the graph can also be used to calculate forces between the nodes. A spring force was used by KK algorithm which is proportional to the graph theoretical distance. For Kamada-Kawai Range Algorithm (KKR), the authors of [25] used weighted graph distance instead of un-weighted graph distance and the range data is incorporated. In network scenario this information is calculated using RSS and time-of-arrival (TOA) information.

If the network size is small and nodes are distributed within a simple region these classic algorithm works well. But these simple force-directed algorithms cannot construct the vertex location if the graph has larger than fifty nodes. [27] Showed that Multi-scale Kamada-Kawai Range Algorithm (MSKKR) [25] utilize both graph adjacency and range information is capable of overcoming this problem.

If the sensor nodes are distributed in a non-convex polygon the algorithms discussed so far yield-unsatisfied result. Finally this article propose a new type of Force-directed algorithm and on top of utilizing graph adjacency and range information, angular information is also integrated in a dead reckoning fashion known as Multi-Scale Dead-Reckoning Algorithm (MSDR) [25].

In our case, the main focus while implementing a force directed algorithm was to assume that each mobile node contained little or no information about its position. The use of simple force directed algorithms with anchor free nodes allows the system to automatically stabilize itself by minimizing the amount of energy within the system. Each nodes adjacent neighbour is attached with a spring with a variable spring constant while on the other hand every node was treated as an electron. If the system detects an unstable topology the program continuously loops until it reaches a stable position.

Another major difference between our objectives and the objective of the graph visualization algorithms such as FR and KK lies in the fact that in those algorithms the locations are relative; i.e. the graph can be rotated or moved around as a whole without any problem because the connectivity between nodes would not change. However this cannot be allowed in a situational awareness application where the absolute node locations are needed.

However throughout the process of minimization each node is associated with a GPS coordinate relative to a default position supplied during program initialization. As a result the physics engine relies on the GPS coordinate system for calculations. As an important side note if the adjacent matrix provided is smaller in size then the previous snapshot the program undergoes reinitializes, as information is lost. If a node becomes disconnected throughout the lifetime of the application the last known position is stored and flagged as inactive. To us it was important to remember these disconnected nodes as mobile mesh networks are very sporadic, chaotic, and random. The program can also be supplied with positions for each node (whether partial or full) which results in those nodes being anchored to their position. These selected nodes only provided electrostatic forces for nodes whether or not they are attached to other nodes with springs. When a node is supplied with location information the system recalculates the average position and realigns the canvas accordingly. By allowing this partial information the system can slowly predict future nodes as they appear. The system built allows the expansion of more information such as signal strength. Systems should be dynamic in a sense where information will or will not be provided on a real time basis. Over time throughout the application lifetime when information is provided the system grows smarter if and only if more information is provided.

Also our proposed approach provides a better balance of attractive and repulsive forces comparing to the FR method in which the attractive force between two nodes is too strong comparing to the repulsive force. Because our objective is to provide an approximation of the real network locations and not merely graph visualization, the balancing or uniformity of graph view which was the objective of the KK and FR algorithms is no longer relevant here, therefore the concept of optimal distance is not used. As our proposed technique is a coarse localization method it has the flexibility to selectively refined by applying the other localization methods on top of it if needed.

Chapter Four: Bandwidth Analysis

Due to the dynamic nature of MANETs, the routing protocol itself could consume a substantial amount of network bandwidth, which would have an impact on the performance of the Tactical network because of the various limitations of nodes (e.g., valuable bandwidth and power will be wasted). Therefore, analyzing network routing overhead in TMANET is very important for achieving required network performance [49] [50] [33]. In our case we are interested in analysing the standard OLSR overhead and as well as the additional overhead for centralized situational awareness.

With the use of OLSR we lose link information because we only use the connectivity as opposed to communication range. If a centralized approach is used, i.e. the topology generator has access to databases on all nodes, and then a complete view of all nodes can be constructed using the 1st- and 2nd-hop neighbour databases only. One problem with relying on OLSR topology databases is that only links that are on the shortest path trees will be discovered. In the centralized visualization approach this would not cause a problem because, assuming that in a typical wireless MANET that nodes always connect to their nearest nodes, a link between a node and its neighbour is included in at least two shortest path trees and therefore, if the central visualizer has access to all databases, it can find all links as well. Furthermore, OLSR TC messages have the option of only reporting the links between an MPR and its selectors, therefore potentially omitting links between non MPR nodes, in such case only a centralized approach that uses 1-hop neighbour databases from all nodes can guarantee the full topology map.

Keeping this issue in mind to evaluate the difference in overhead between the distributed and centralized approach we generate a custom message named *NIM* message that will be broadcasted throughout the network with its 1-hop neighbour information using OLSR default forwarding algorithm. Though it will generate network overhead but it will allow us to discover network topology that is not possible by using OLSR information.

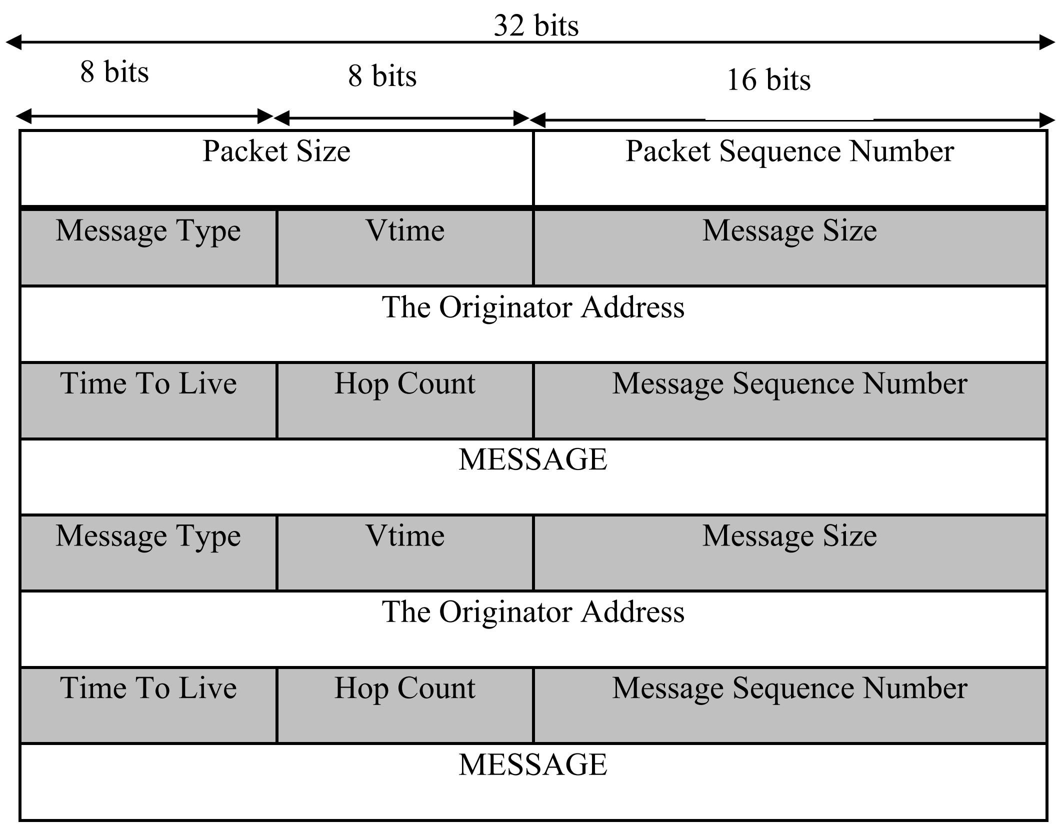


Figure 4-1: OLSR Packet Lay Out.

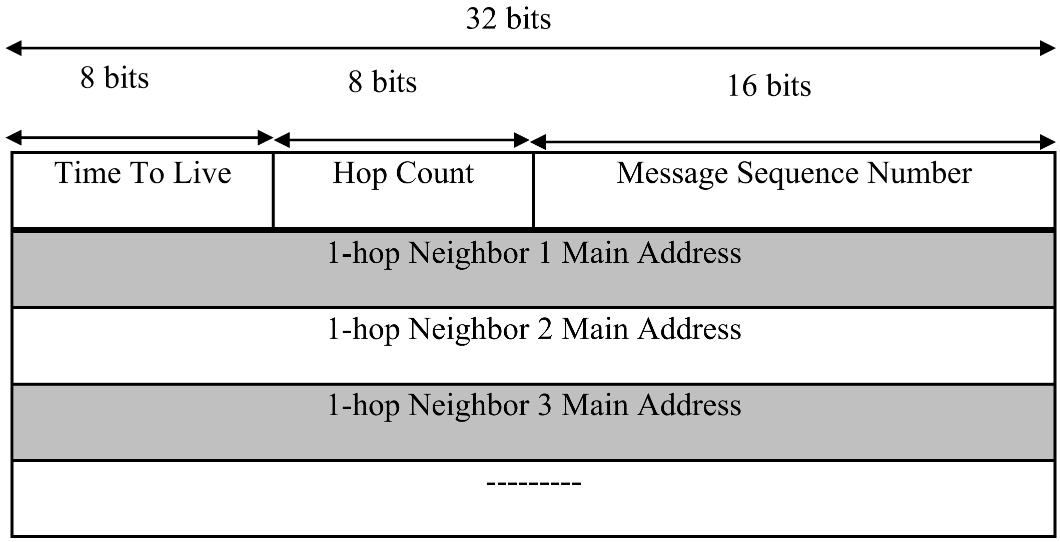


Figure 4-2: Neighbour Information (NIM) Message Format.

In [30] [49] [51], the authors analyzed the network overhead and performance analytically while various routing protocols performance are appraised and compared with NS-2 simulation in [52]. Also in [53], an analytical model for calculating standard OLSR overhead at the IP layer is proposed. In our research we are using this model as a base enhancing it to evaluate the overhead difference between centralized and distributed approach including the overhead for NIM messages. We also calculate the network overhead using the NS-3 simulator to compare our analytical results. For our model we initialize a list of parameters divided into OLSR configuration parameters and network scenario parameters from the overhead point of view [53].

4.1 **Definition of Parameters for analytical Model**

4.1.1 OLSR Protocol Parameters

- H_{Hello}: OLSR *Hello* message header size.
- H_{TC} : OLSR Topology Control message header size.
- H_{NIM} : OLSR Neighbour Information message header size.
- H_{Msg}: OLSR message header size.
- H_{Pkt}: Header for OLSR packet.
- Frq_{Hello}: Frequency for transmitting OLSR *Hello* messages.
- Frq_{TC} : Frequency for broadcasting OLSR *TC* messages.
- Frq_{NIM} : Frequency for broadcasting OLSR *NIM* messages.

4.1.2 Parameters for Network Scenario

- N: The total number of nodes within the network.
- M: The total number of MPR nodes within the network.
- $Neigh_{avg}$: Average nodal degree or number of neighbours' node per node.

  According to OLSR several nodes can select one node as an MPR and the selector node is known as MPR selector [20]. Therefore, we define a parameter MS_{avg} to denote the average number of MPR selector for each MPR.Depending on the topology of the network there may be some nodes, which are not selected by any of their neighbour's node as an MPR node. In

- Figure 4-3 (c), m_0 is not selected by any of its neighbour as an MPR node. So for calculating the average number of this type of node within the network we define another parameter $M_{0_{avg}}$.

- $TC_{Tx_{avg}}$: The average number of retransmission per *TC* message combining the first transmission.
- $TC_{Rx_{avg}}$: The average number of TC messages received by a node within a period $\frac{1}{Frq_{TC}}$.
- $NIM_{Tx_{avg}}$: The average number of retransmission per *NIM* message combining the first transmission.
- $NIM_{Rx_{avg}}$: The average number of *NIM* messages received by a node within a period $\frac{1}{Frq_{NIM}}$.

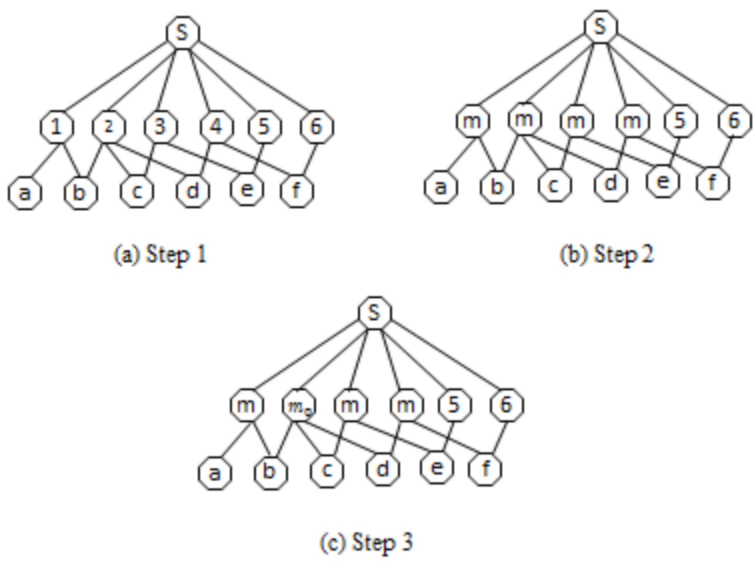


Figure 4-3: MPR selection procedure.

4.2 **Process for Obtaining Parameters**

The parameters for OLSR protocol are available in [20] and the parameters for network scenario [53] can be found by following some easy steps mentioned below:

- For each single node i, the set of its neighbour nodes $S(i)$ only contains those nodes that are within one hop range of itself.

So, $Neigh_{avg}(i) = |S(i)|$ and $Neigh_{avg} = \frac{1}{N}\sum_{i=1}^{N} Neigh_{avg}(i)$ (16)

- For each single node i, the set of its MPR nodes $S_M\ (i)$ will be calculated according to OLSR MPR node selection algorithm.

So, $M = \sum_{i=1}^{N} \alpha_i$ (17)

Where $\alpha_i = 1$, if $\exists\ j,\ i \in S_M\ (j)$ and 0 otherwise.

And $Avg_{MS} = \frac{1}{M}\sum_{i=1}^{N} |S_M(i)|$ (18)

Also $M_{0_{avg}}(i) = \sum_{j \in S(i)} \beta_j$ (19)

Where $\beta_j = 1$, if $\forall k, j \notin\ S_M\ (k)$ and 0 otherwise.

So, $M_{0_{avg}} = \frac{1}{N}\sum_{i=1}^{N} M_{0_{avg}}(i)$ (20)

- If node i is acting as an MPR itself than the list of its own MPR nodes $L_M(i)$ will be responsible for retransmitting the TC messages that is originated from it based on the rule specified in OLSR [20], i.e. if the copy of the TC message is received the first time from an MPR selector than it will be retransmitted otherwise it will be discarded.

So, $TC_{Tx_{avg}}\ (i) = L_M(i)$ and $TC_{Tx_{avg}} = \frac{1}{M}\sum_{i=1}^{N} TC_{Tx_{avg}}(i)$ (21)

As we know from [20] that within an OLSR network MPR nodes are only responsible for handling TC messages and as a result the total number of nodes participating in the *TC* retransmission is exactly M and also a *TC* message never be retransmitted twice by an MPR. So, within a period $TC_{Tx_{avg}}$ is also the average number of topology control messages that are generated and transmitted by an MPR node and $M\,TC_{Tx_{avg}}$ will be the total number of TC transmission including the retransmission of the same message within a whole network.

- If each node receives a TC message, $TC_{Rx}\ then$

$$TC_{Rx_{avg}} = \frac{1}{N}\sum_{i=1}^{N} i.TC_{Rx} \quad (22)$$

- For each single node i, the set of its neighbour nodes $S(i)$ will be responsible for retransmitting the neighbour information message originated from it.

So, $NIM_{Tx_{avg}}\,(i) = S(i)$ and $NIM_{Tx_{avg}} = \frac{1}{N}\sum_{i=1}^{N} NIM_{Tx_{avg}}(i)$ (23)

- If each node i receives a *NIM* message, NIM_{Rx} then

$$NIM_{Rx_{avg}} = \frac{1}{N}\sum_{i=1}^{N} i.NIM_{Rx} \quad (24)$$

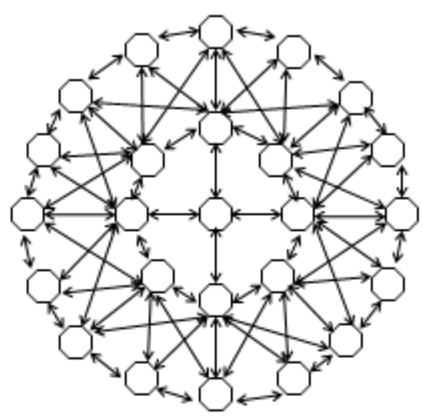


Figure 4-4: Flooding Mechanism for the NIM message.

4.3 **Analytical Computation of the Network Overhead**

Based on the average consumption of bandwidth by each node for exchanging OLSR messages within the network, the OLSR protocol overhead is calculated in bits per second or bytes per second. The bandwidth consumed for sending and receiving *Hello*, *TC* and *NIM* messages can be denoted as: Oh_{SHello}, Oh_{RHello}, Oh_{STC} , Oh_{RTC}, Oh_{SNIM}, Oh_{RNIM}.

We also denote sz_{Hello} and sz_{TC} as the average *Hello* and *TC* packet size as well as sz_{NIM} as the average *NIM* packet size. As there is only one OLSR message in each OLSR packet [20]:

$$Oh_{RHello} = Frq_{Hello}.Neigh_{avg}.sz_{Hello} \qquad (25)$$

$$Oh_{SHello} = Frq_{Hello}.sz_{Hello} \qquad (26)$$

$$Oh_{RTC} = Frq_{TC}.TC_{Rx_{avg}}.sz_{TC} \qquad (27)$$

As it discussed in the previous section, within a period $\frac{1}{Frq_{TC}}$, $N\,TC_{Tx_{avg}}$ is the average number of topology control messages that are generated and transmitted by an MPR node. Thus, for each node:

$$Oh_{STC} = Frq_{TC}.\frac{M\,TC_{Tx_{avg}}}{N}.sz_{TC} \qquad (28)$$

$$Oh_{SNIM} = Frq_{NIM}.NIM_{Tx_{avg}}.sz_{NIM} \qquad (29)$$

$$Oh_{RNIM} = Frq_{NIM}.NIM_{Rx_{avg}}.sz_{NIM} \qquad (30)$$

Now, the total overhead of OLSR protocol will be the sum of transmission and reception of OLSR packets throughout the network within a period of time.

Finally,

$$Oh_{OLSR} = Oh_{SHello} + Oh_{RHello} + Oh_{STC} + Oh_{RTC} + Oh_{SNIM} + Oh_{RNIM}$$

$$= Frq_{Hello}\left(Neigh_{avg} + 1\right).sz_{Hello} + Frq_{TC}\left(TC_{Rx_{avg}} + \frac{M\,TC_{Tx_{avg}}}{N}\right).sz_{TC} +$$

$$Frq_{NIM}\left(NIM_{Rx_{avg}} + NIM_{Tx_{avg}}\right).sz_{NIM} \quad (31)$$

In order to calculate the size of *Hello* message, *TC* and *NIM* message, we also have to consider the size of address (sz_{Addr}) and the size of link codes (sz_{Lc}). The link code mainly specifies the type of link and advertised in each *Hello* message and it can be symmetric, asymmetric, multipoint relay or lost. After coming to a stable state, the network should have only two kinds of link type: symmetric and MPR links. The size of the packet than be calculated as the following equation.

$$sz_{Hello} = Neigh_{avg}.sz_{Addr} + 2.sz_{Lc} + H_{Hello} + H_{Msg} + H_{Pkt} + H_{IP} \quad (32)$$

$$sz_{TC} = MS_{avg}.sz_{Addr} + H_{TC} + H_{Msg} + H_{Pkt} + H_{IP} \quad (33)$$

$$sz_{NIM} = Neigh_{avg}.sz_{Addr} + H_{NIM} + H_{Msg} + H_{Pkt} + H_{IP} \quad (34)$$

The values for different OLSR parameters are tabulated below:

Table 4-1: Size of OLSR parameters and components.

Parameter	**Value (bytes)**
$\boldsymbol{H_{Hello}}$	4
$\boldsymbol{H_{TC}}$	4
$\boldsymbol{H_{Msg}}$	12
$\boldsymbol{H_{NIM}}$	4
$\boldsymbol{H_{Pkt}}$	4
$\boldsymbol{H_{IP}}$	20
$\boldsymbol{sz_{Addr}}$	4 (IPv4)
$\boldsymbol{sz_{Lc}}$	4

For corroborating our analytical model, we compare our analytical result with NS-3 [53] simulation result with different network scenario. We also examine the impact of node density, mobility, NIM, TC and Hello messages on bandwidth overhead for centralized and nodal_view and they are presented briefly in the Performance Analysis chapter.

Chapter Five: Performance Evaluation

This chapter describes the framework used to simulate the test scenario and presents the simulation results for evaluating the performance of our proposed topology discovery approach in section 5.2. In section 5.3, comparing with simulation results followed by the Section 5.4 where the performance evaluation of our proposed approximate localization technique is presented validates our analytical model for network bandwidth analysis.

5.1 **Simulation Environment**

5.1.1 Network simulator-3

According to NS-3 working group-

ns-3 is a discrete-event network simulator, targeted primarily for research and educational use. ns-3 is free software, licensed under the GNU GPLv2 licences, and is publicly available for research, development, and use [54].

For the installing of ns-3 (Version 3.10), we used the Linux operating system and version was Ubuntu 9.10. As it was written in C++ all of our simulation is scenario is developed using C++.

5.1.2 Simulation Methodology and Parameters

As mentioned, a simulation environment was set up for implementing MANET using NS-3, which is a discrete-event network simulator. We created different scenarios for network sizes of 5, 10, 15, 20, 35 and 50 nodes, and each scenario were simulated with different mobility models and then with constant position (non-mobile) for comparison. Also IEEE 802.11a is used as wireless MAC Layer, where each node was capable of establishing ad-hoc connection with all other nodes within its range. As a routing protocol a table driven proactive OLSR routing protocol was used. For the wireless channel we use Yans-wifi channel propagation model [55]. In our simulation, a constant speed propagation delay model and Friss propagation loss model were used to represent the propagation delay and the loss of signal power in a free space as distance increases. For modeling 802.11a physical layer we also defined the Tx power = -0.1615 dBm and the data rate was set to 4 kb/s. For various mobility models, a closed area of 500 x 500 sq. meters was assumed where the nodes moved randomly within the area. In non-mobile case, the nodes were placed on a grid within the same area.

We used NS-3 logging feature [56] to log all OLSR events. A custom script was created to parse the log file and re-generate events related to the topology database in real time and to send them to a topology calculation module. In order to create the full_view, OLSR events for all nodes were processed while creating nodal_view required only events on a node under consideration. The script checked the log file for events that contain information about active links between a node and its 1st- and 2nd-hop neighbours, as well as the last hop before a destination.

The parameters we used in the simulation are tabulated in following table:

Table 5-1: simulation Parameters

Parameter	Value
Size of area	500 x 500 m
Number of nodes	5, 10, 20, 35 and 50
Wireless Model	IEEE 802.11 a
Transmission range	150 m
Mobility model	Random direction 2D, Random way point, Random walk 2D, Gauss Markov and Constant Position (Static)
Node speed	Random
Delay model	Constant Speed Propagation Delay and Friss Propagation loss Model
Simulation time	100 seconds

5.2 **Accuracy and Connectivity**

In this section we present the results for our topology discovery approach. As we already discussed before OLSR protocol periodically declares 1-hop, 2-hop and destination node information for each and every node within a network. The total number of active links at an instance of time was calculated by adding up all the links based on this information. We assumed symmetric network links between nodes. We then generated adjacency matrices based on link information for full view topology and each partial view topology.

If all links are bidirectional, the adjacency matrix in this case will be symmetric. The adjacency matrix is updated periodically as OLSR updates its tables.

At each instant once the adjacency matrices were built for each nodal_view (distributed view), and combined to create the full_view (centralized view). We compared each nodal_view adjacency matrix with full_view adjacency matrix to determine the number of undiscovered links and finally the accuracy of nodal_view topology discovery.

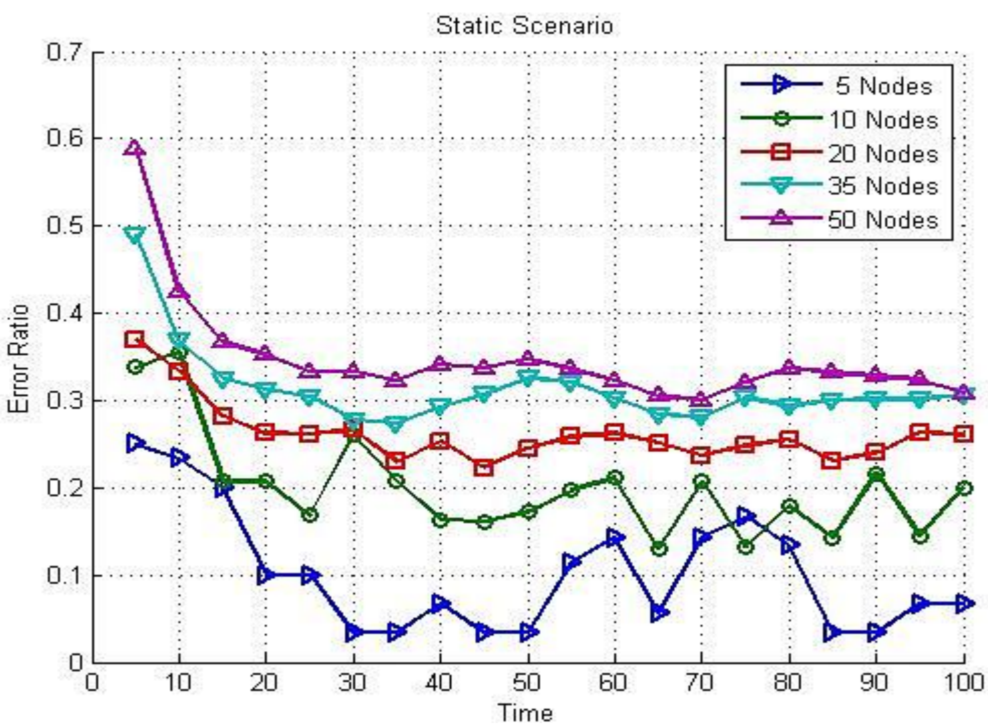


Figure 5-1: The average error rate for different network size for the static scenario.

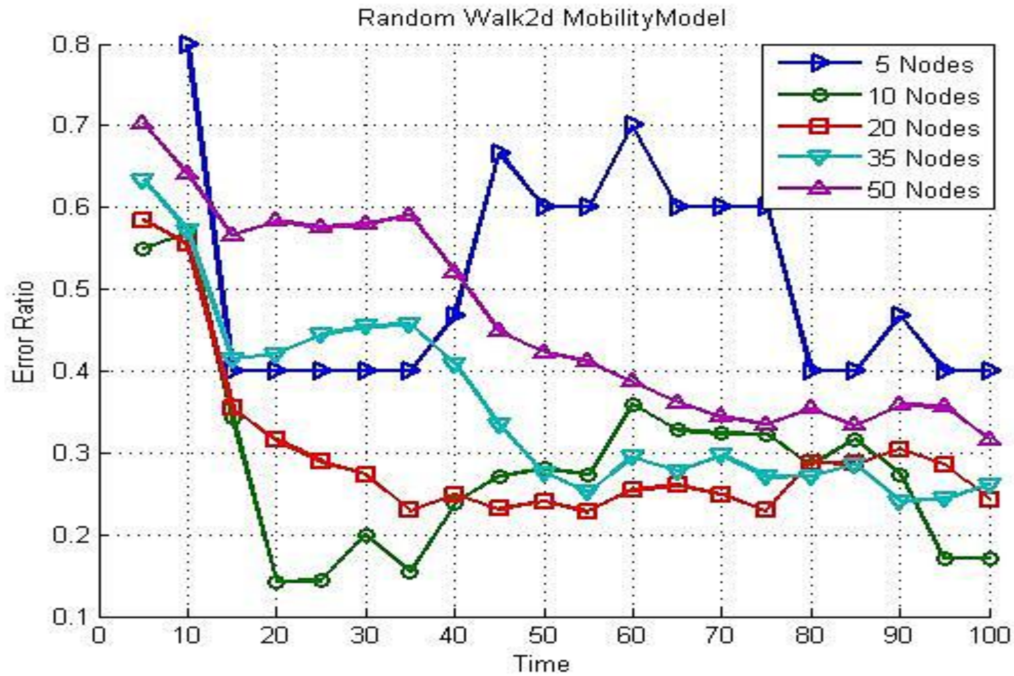


Figure 5-2: The average error rate for different network size with Random Walk 2d mobility model.

Figure 5-1, Figure 5-2Figure 5-3Figure 5-4 and Figure 5-5 shows the error rate of topology discovery for different network sizes with and without mobility model. Each time step identifies a visualizer update interval, set to 5 seconds. Each simulation ran for 100 seconds. In both mobile and non-mobile cases the number of undiscovered link in nodal views (in comparison to the full view) is calculated at an instant of time and presented as error rate. Error rates in Figure 5-1Figure 5-2Figure 5-3Figure 5-4 and Figure 5-5 have been defined as Eq. (35), where -

N_{UL} = The total number of undiscovered links

N_{ALF_V} = The total number of active links at the full_view topology

N_{ALN_V} = The total number of active links at the nodal_view topology

$$\text{Error Rate (ER)} = \frac{N_{UL}}{N_{ALF\_V}} = \frac{N_{ALF\_V} - N_{ALN\_V}}{N_{ALF\_V}} \qquad (35)$$

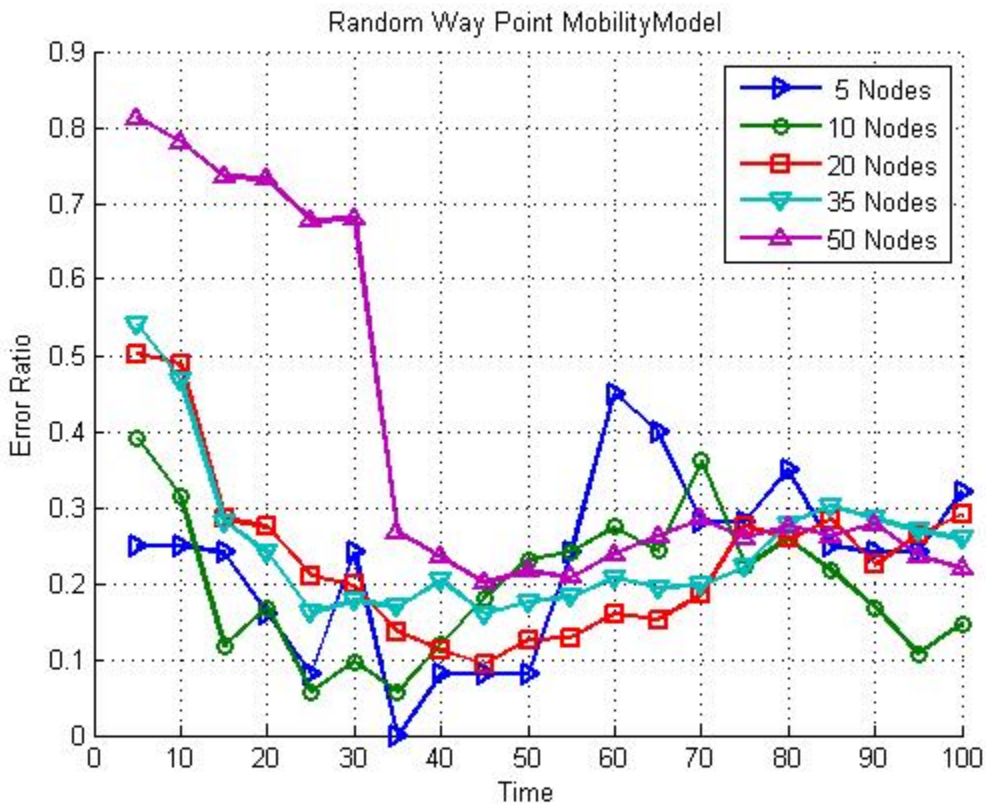


Figure 5-3: The average error rate for different network size with Random Waypoint Mobility Model.

We note that in addition to mobility, there is also a possibility of link loss due to lower signal strength resulting from environment noise levels; therefore even in non-mobile scenario the error rate could increase in time. In the mobile scenario, however, mobility is the main factor in the increase in the number of undiscovered links. Being a proactive routing protocol, OLSR uses a shortest path algorithm to calculate a route instead of flooding mechanism and it continuously updates route information, which results in continuous change of link status among nodes. As a result undiscovered links increase with network size.

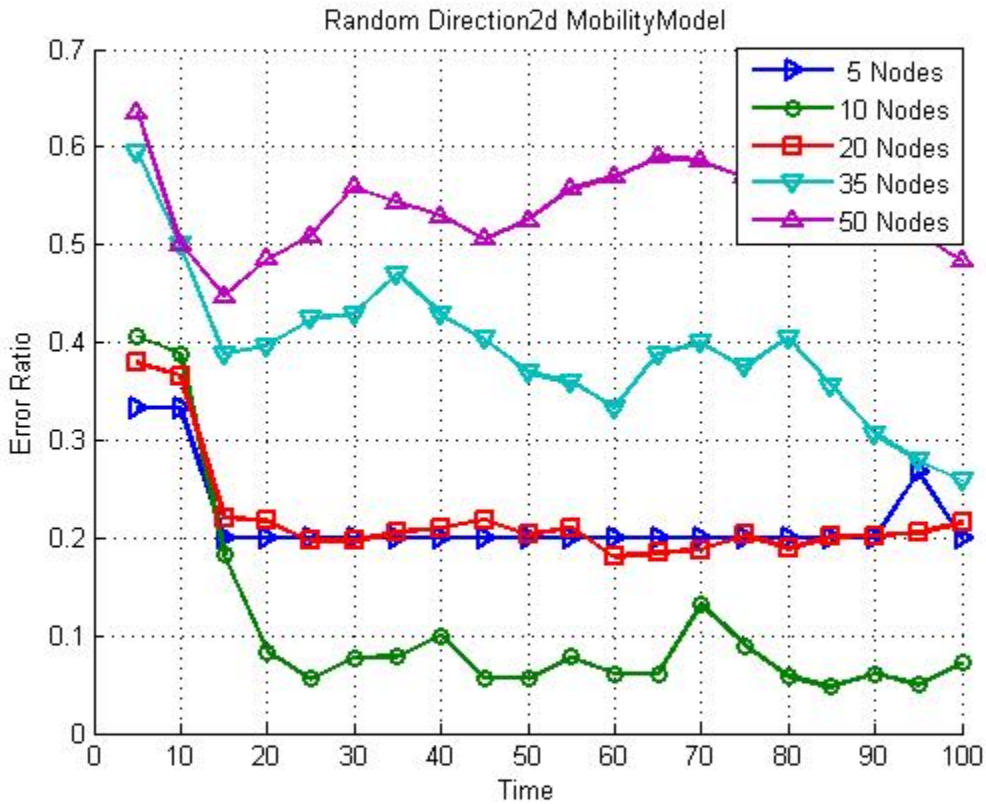


Figure 5-4: Average error rate for different network size with Random Direction2d Mobility Model.

At the first stage of topology discovery the nodal view is rather incomplete for both static and mobile scenarios. Over time, each node discovers more information about network topology. However, full stability is never attained because topology is constantly changing (either because of mobility or change in signal strength). In our simulation, a random expiration time was generated for each active link and after the expiration of this entry; it sensed the link again to check its status. As shown in Figure 5-1Figure 5-2Figure 5-3Figure 5-4 and Figure 5-5 the error rates in the static and mobile scenarios started around 80% and dropped to the 8%-35% range with occasional spikes depending on the mobility models. In the static scenario, the error rates quickly stabilized below 35%

(relating to the links not on the node’s shortest path tree) with occasional increases due to noise and weak signal strength.

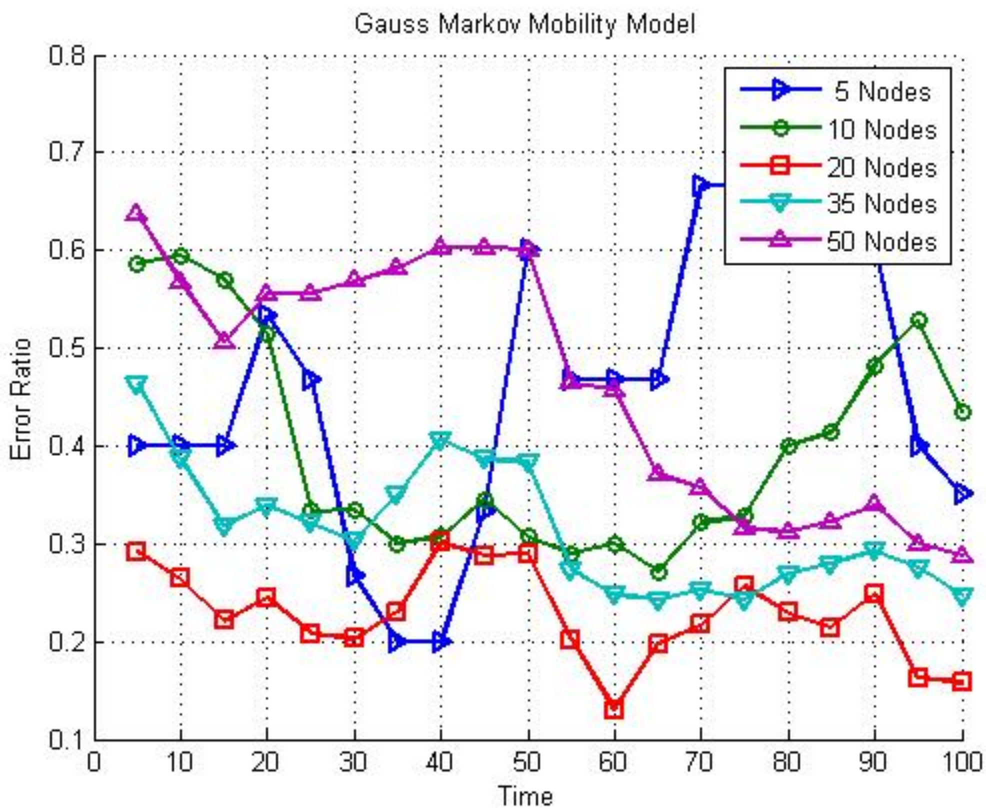


Figure 5-5: The average error rate for different network size with Gauss Markov Mobility Model.

5.3 **Bandwidth Analysis**

In this section, we reported the overhead generated by the OLSR as a function of messages sends and received throughout the network to construct the full view of the network topology. As we discussed in chapter four, the overhead caused by the *NIM* message is calculated and compared with the bandwidth consumption for *HELLO* and *TC* messages. The overhead generated by OLSR is computed as the number of bytes at the IP layer. We use the same simulation environment mentioned in section 5.1.2. Figure 5-6, 5-

7, 5-8, 5-9 and Figure 5-10 illustrate the overhead generated by HELLO, NIM and TC messages.

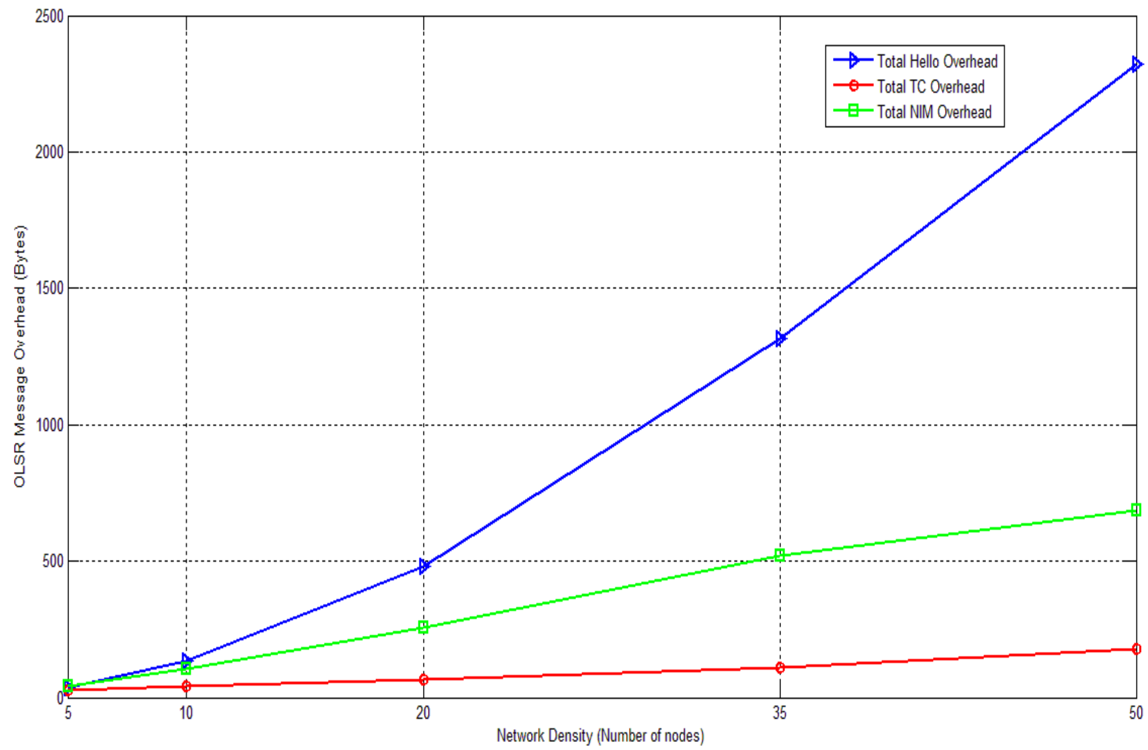


Figure 5-6: Overhead generated by OLSR at static scenario.

Regardless the mobility pattern and network density the overhead generated by the OLSR due to *NIM* messages is always higher than *TC* messages and lower than *HELLO* messages and the reason behind this is that only the MPR nodes are responsible for forwarding *TC* messages and those are only a subset of nodes within the network. But in case of *NIM* messages all nodes within the network will forward the *NIM* messages.

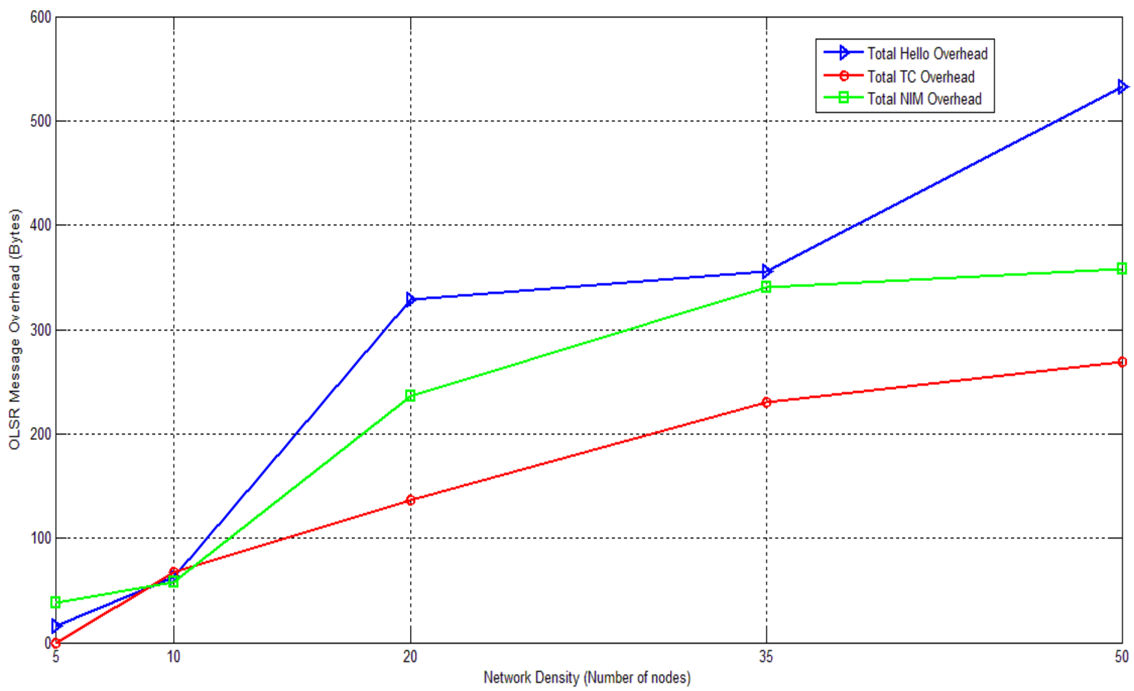


Figure 5-7: Overhead generated by OLSR with Random Walk 2D mobility model.

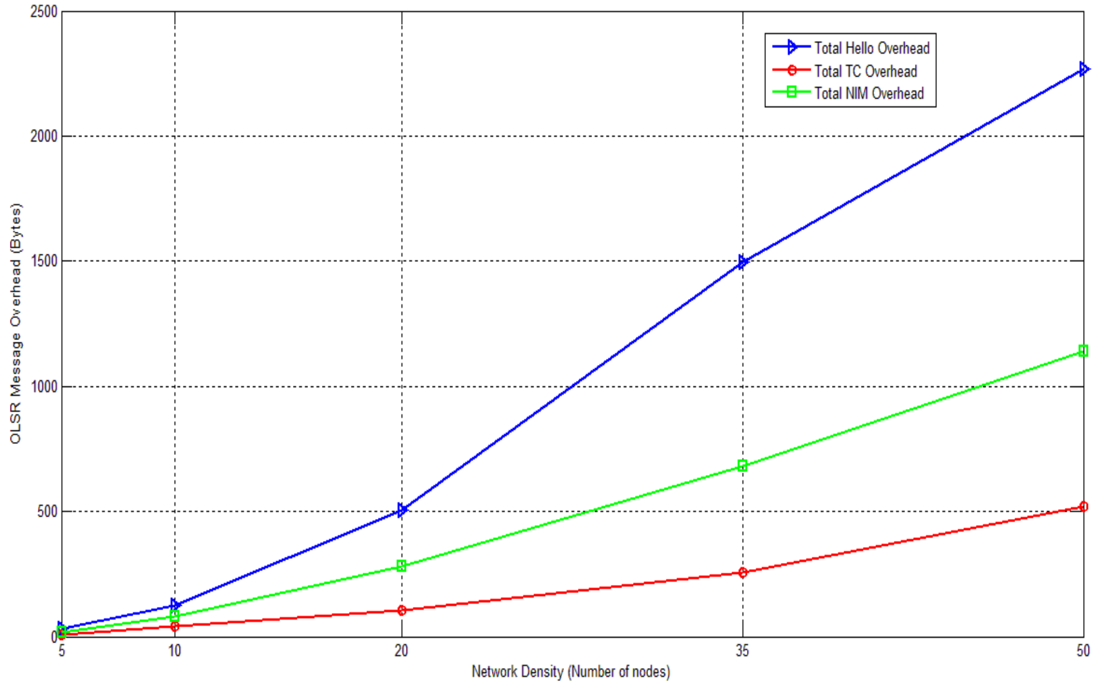


Figure 5-8: Overhead generated by OLSR with Random Waypoint mobility model**.**

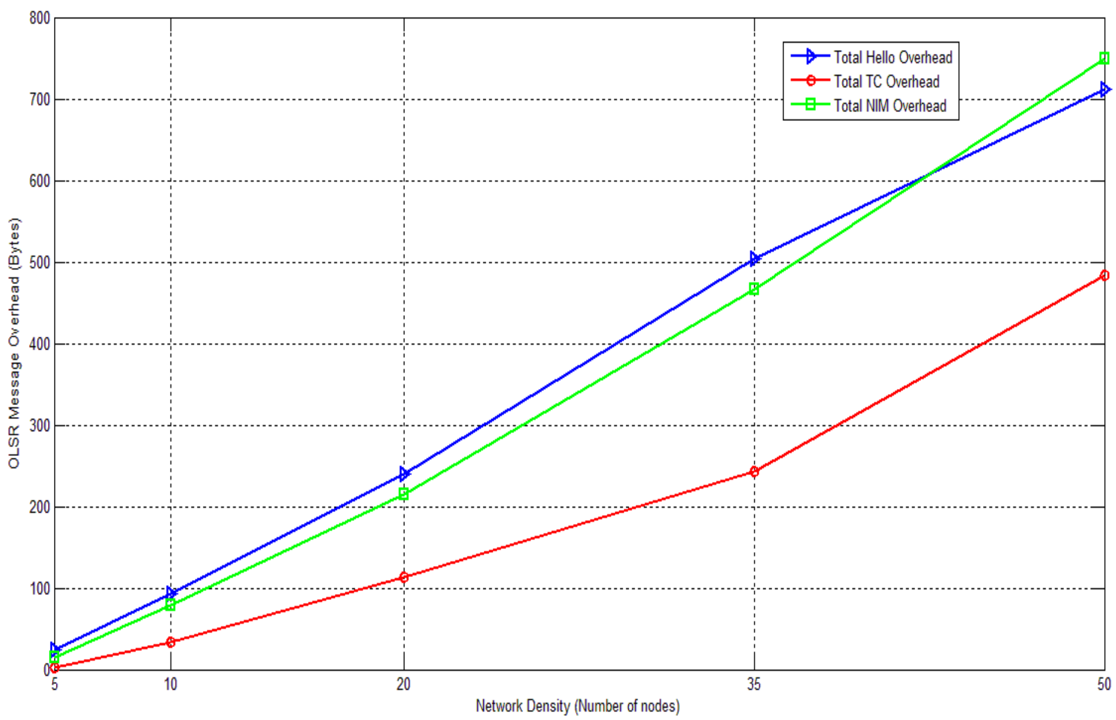


Figure 5-9: Overhead generated by OLSR with Random Direction 2D mobility model.

The bandwidth consumption for the NIM message is almost twice than the bandwidth consumed by TC messages for any network density with any mobility model and as the network size increase the overhead generated by OLSR for the NIM message also increase.

In a centralized visualization approach if we want to have the full view of the network topology and to create that if we want to use the link information from NIM messages on top of TC and HELLO messages there will be an additional 67 % to 75% network overhead.

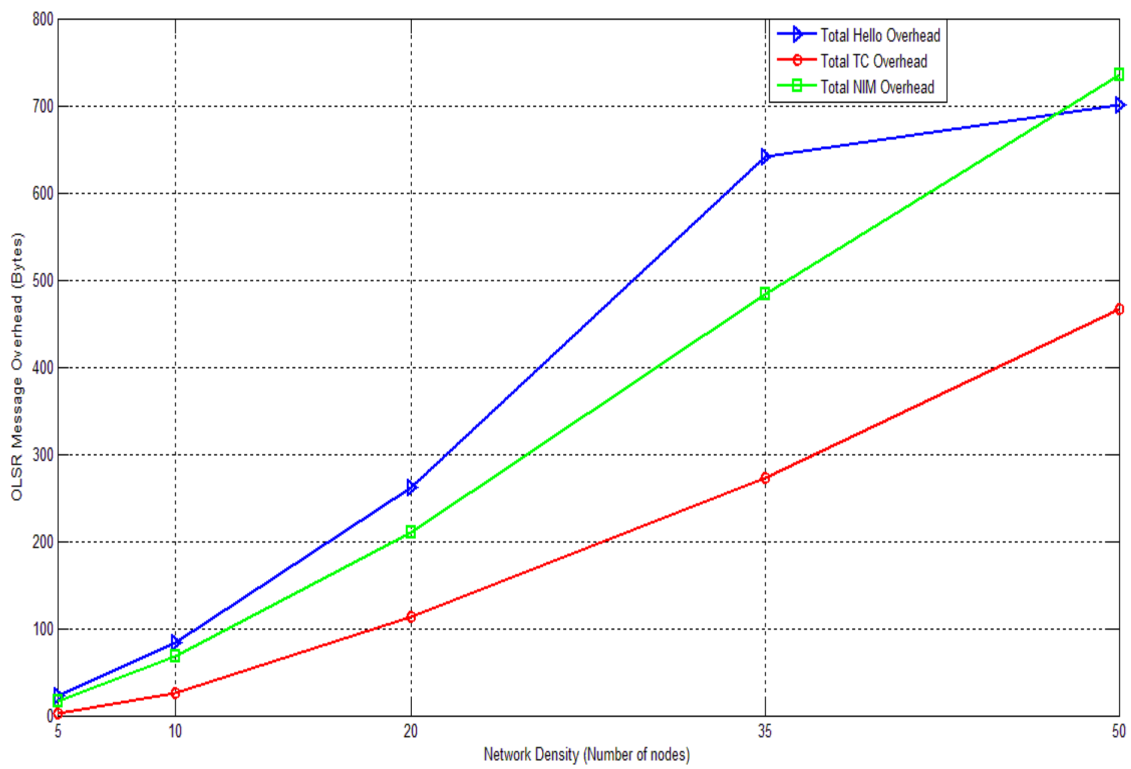


Figure 5-10: Overhead generated by OLSR with the Gauss Markov mobility model.

5.3.1 Model Validation

Our proposed analytical model for bandwidth analysis is validated with simulation result from NS-3. For our analytical model and NS-3 simulation we use the same scenarios. Figure 5-11, 5-12, 5-13, 5-14 and Figure 5-15 shows that our analytical model attains a good accuracy compared to NS-3 simulation, regardless the network size and nodes moving pattern. The error rate between our proposed analytical model and NS-3 simulation is less than 6%.

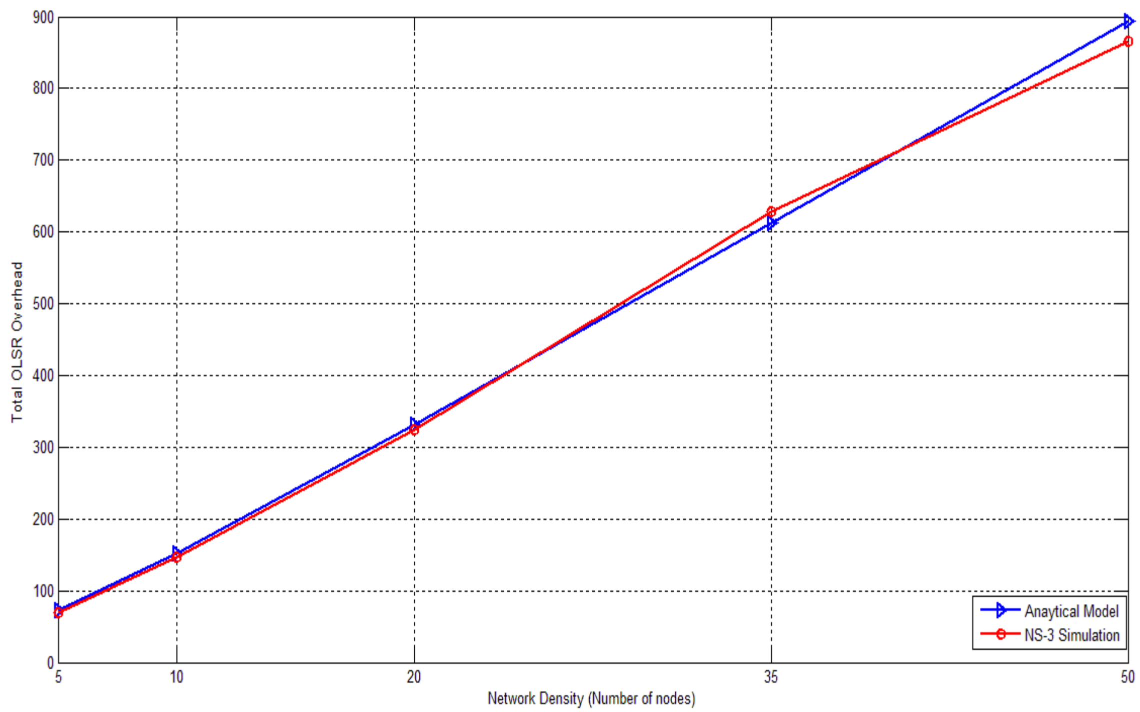


Figure 5-11: Validation of the analytical model with static scenario.

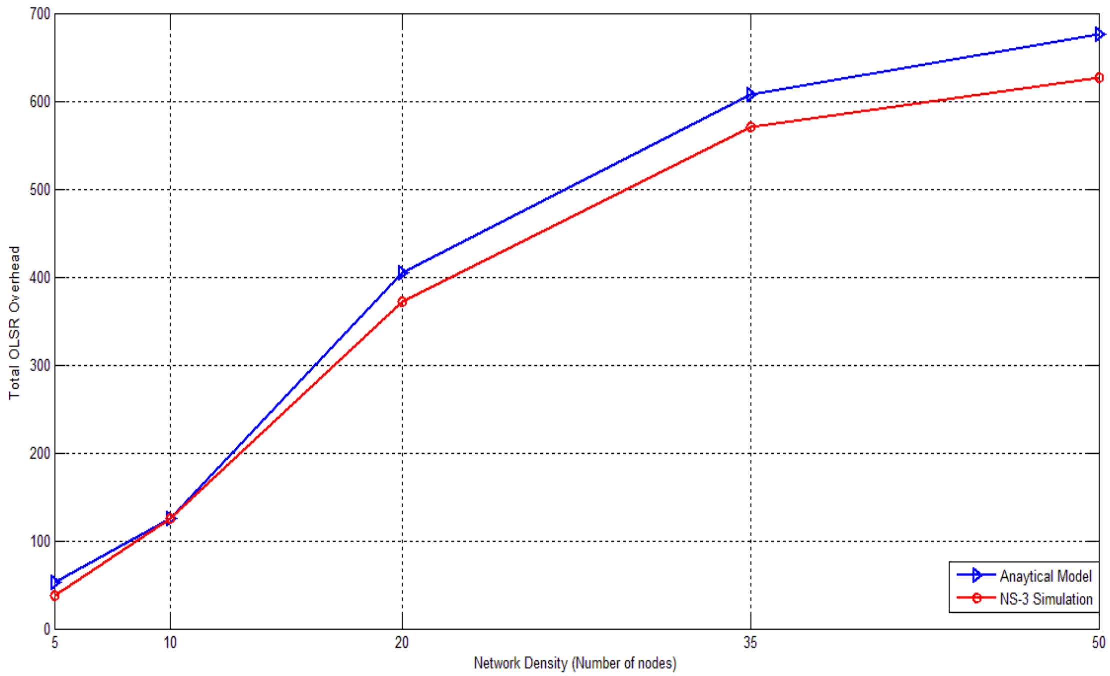


Figure 5-12: Validation of the analytical model with Random Walk 2D mobility Model.

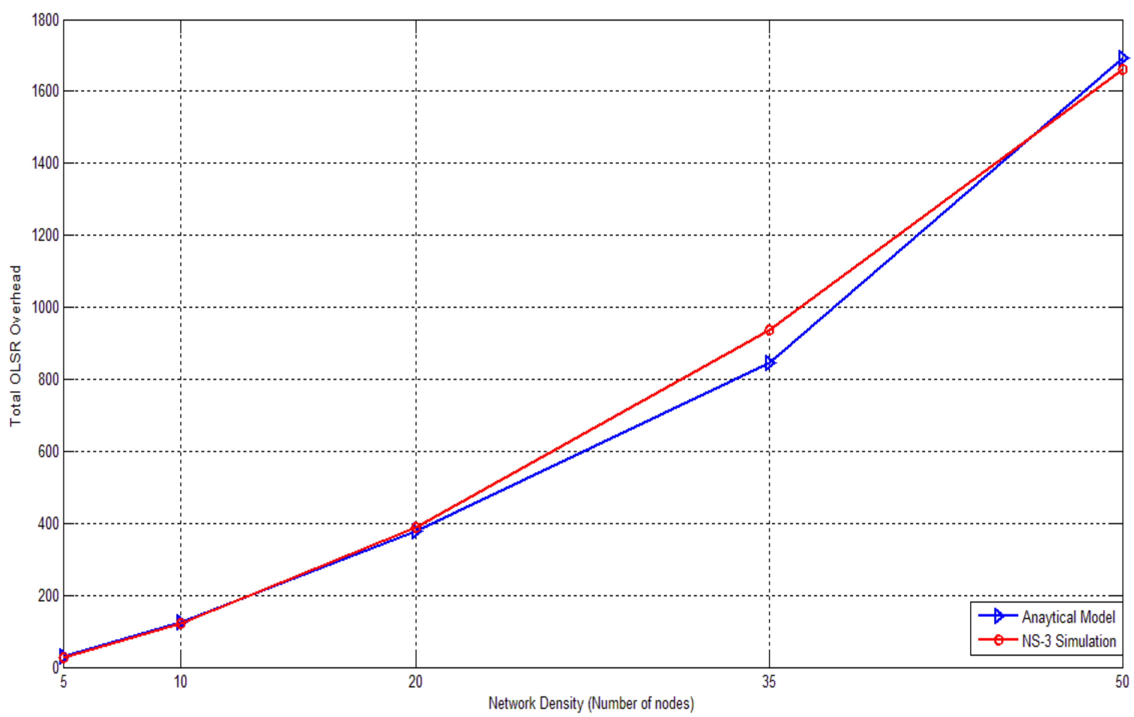


Figure 5-13: Validation of the analytical model with Random Waypoint mobility Model.

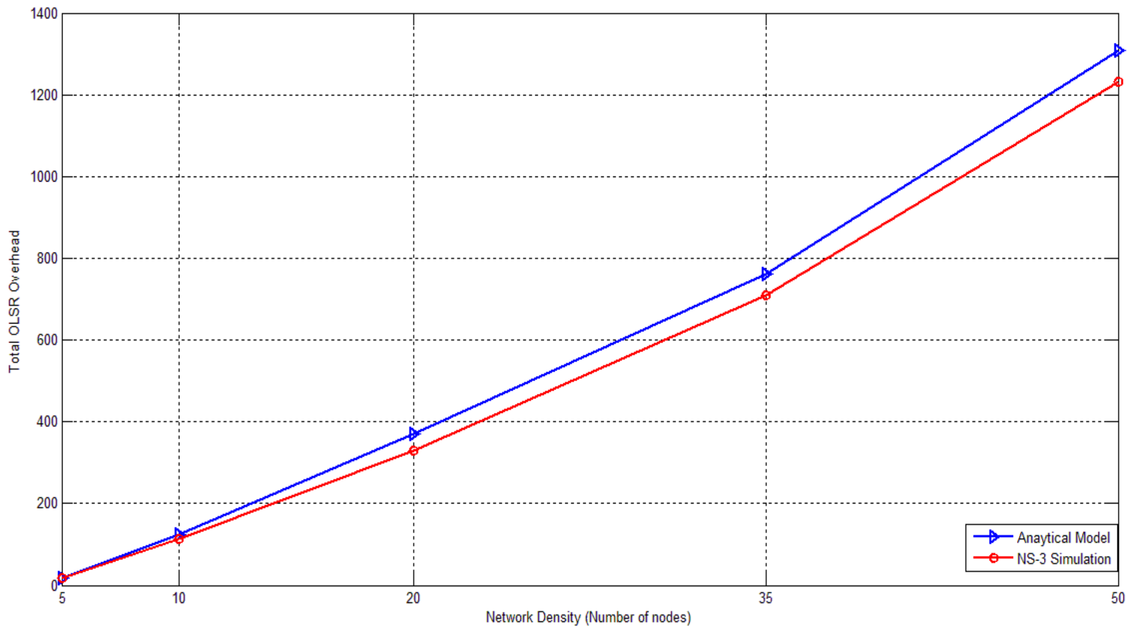


Figure 5-14: Validation of the analytical model with Random Direction 2D mobility Model.

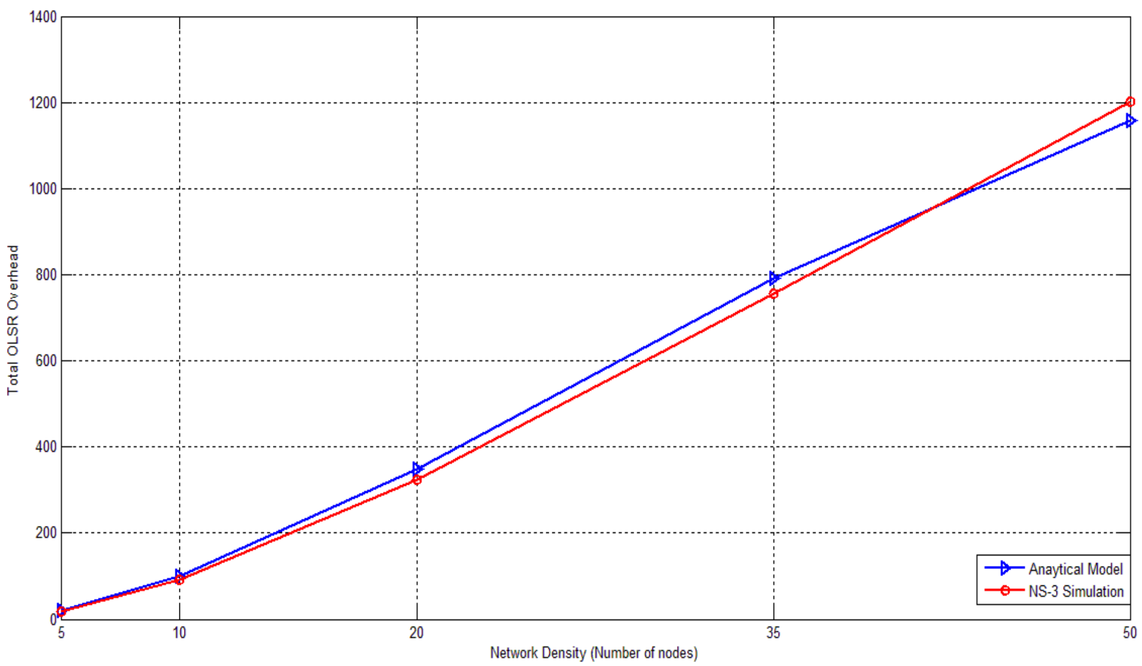


Figure 5-15: Validation of the analytical model with Gauss Markov mobility Model.

For network size more than or equal to 20 nodes, the network overhead is overestimated by the analytical model due to the fact that collisions between packets are not considered in the analytical model, whereas in real world scenario or in our case in NS-3 simulations a transmitted packet can encounter a collision and as a result may lost and lo longer forwarded.

5.4 **Coarse Localization**

In order to evaluate the proposed approach described in chapter three to calculate the approximate location of MANET nodes, a variety of simulation environments are created to cover a wide range of network as mentioned in section 5.1.2. The topology information of the network is fed into the visualizer (Called NetViz) as an input for force directed algorithm to calculate the approximate position of the nodes. For better accuracy we can also provide information regarding any GPS enabled (GE) node or any landmark coordinates. The algorithm checked the input topology file for events that contain information about active links between a node and its 1st- and 2nd-hop neighbours, as well as the last hop node before a destination and the destination node and as well as for any GPS or landmark information.

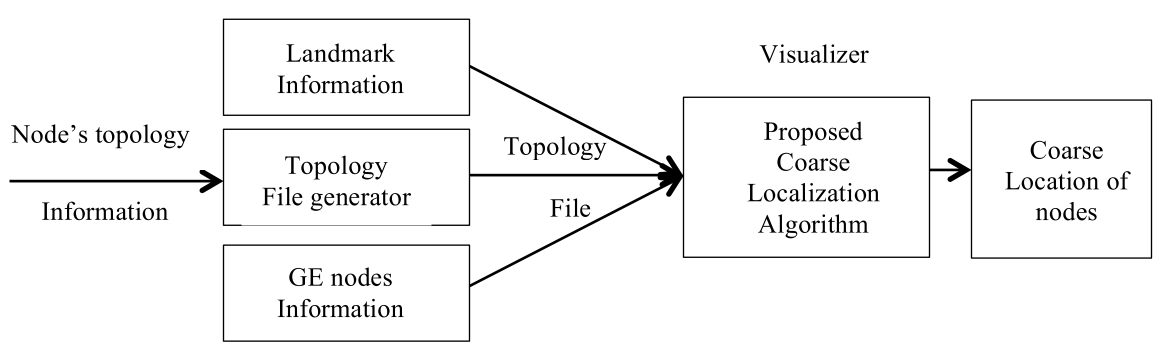


Figure 5-16: Coarse Localization Process

At each instant the proposed force directed algorithm calculated the position information in X and Y coordinate format and save as a log file and compare the log with our simulated position coordinates to calculate the X and Y coordinates deviation; i.e. if (X_o, Y_o) is the original position of a node N_i and (X_c, Y_c) is the calculated position by FDA at any instant of time than the deviation is calculated using Eq. (36) and Eq. (37).

$$\text{Deviation of X (m)} = X_o - X_C \quad (36)$$

$$\text{Deviation of Y (m)} = Y_o - Y_C \quad (37)$$

We also calculate the distance between the nodes using Eq. (38) to determine the accuracy of our coarse localization approach that is the average difference in the computed location to actual location.

$$Distance\ (d) = \sqrt{(X_0 - X_c)^2 + (Y_0 - Y_c)^2} \qquad (38)$$

To determine the accuracy of our proposed approach we tested our FDA algorithm with three different scenarios. First of all we assume that all the nodes are GPS enabled and for the second scenario we assume that only a small portion of the nodes have GPS information and finally we provide landmark information in addition to GPs information. As it shows in Figure 5-18, a screen shot from visualizer where an example scenario is shown where all the nodes have GPS information and N10 is a moving node.

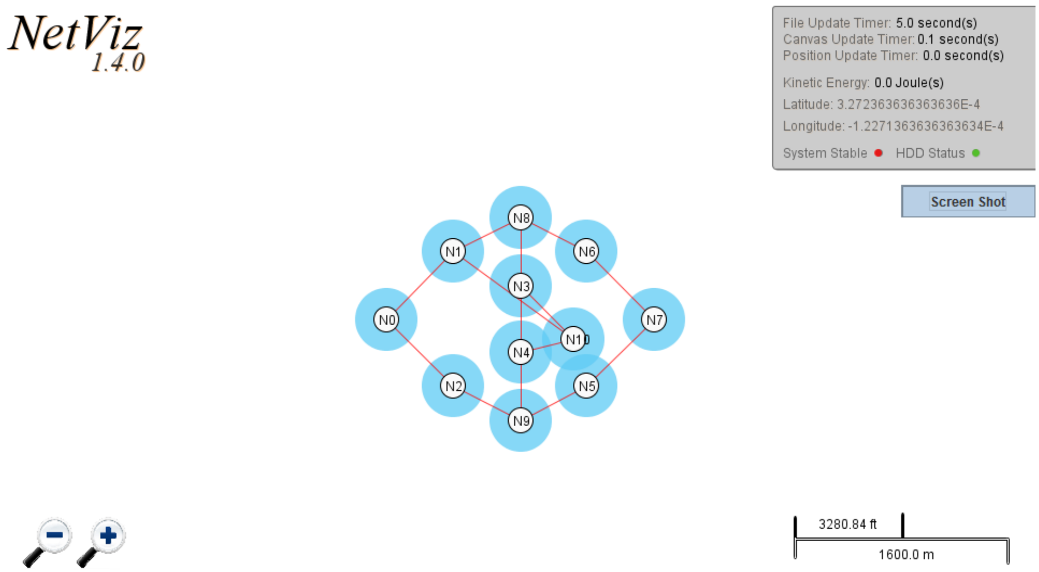


Figure 5-17: Net Viz Visualizer (Example Scenario with full GPS information).

Also in Figure 5-18 same scenario is shown where N8, N9 are the GPS enabled node and N0, N7 are considered as landmarks and rest of the nodes do not have any position information. It also shows the displacement of the nodes (Without location information) as the attractive and repulsive forces act on them based on our algorithm.

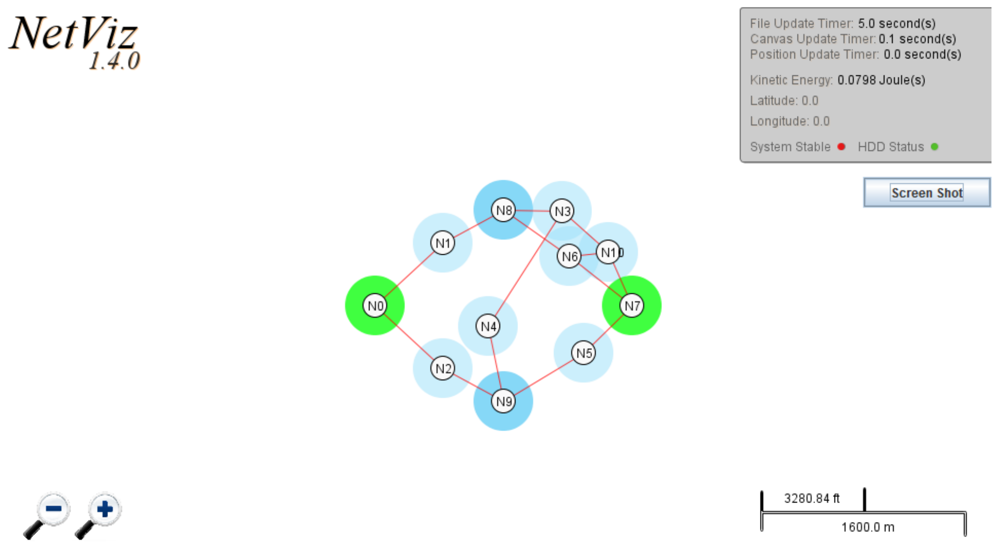


Figure 5-18: NetViz visualizer (Example Scenario with partial GPS and Landmark information)

Figure 5-19, 5-20, 5-21, 5-22 and Figure 5-23 shows the deviation of X and Y coordinates from the original position where nodes are static and moving with Random walk 2D, Random direction 2D, Random way point and Gauss Markov mobility model respectively for three different scenarios mentioned early. With fully GPS enabled the deviation from original position is zero in all cases. After providing 25% of nodes with GPS information the deviation rises up to 400 m. But as we see from the figures that this amount vary for different mobility models as the pattern of the nodes movement is different resulting the change in topology of the network. If we add 10% landmark information the amount of deviation decreases significantly up to 50 m. The problem associated with relaying only GPs enabled node is that sometimes it may become dead, i.e. the node can be disconnected from all other node and its position information cannot be used by other nodes as a reference to calculate its own position.

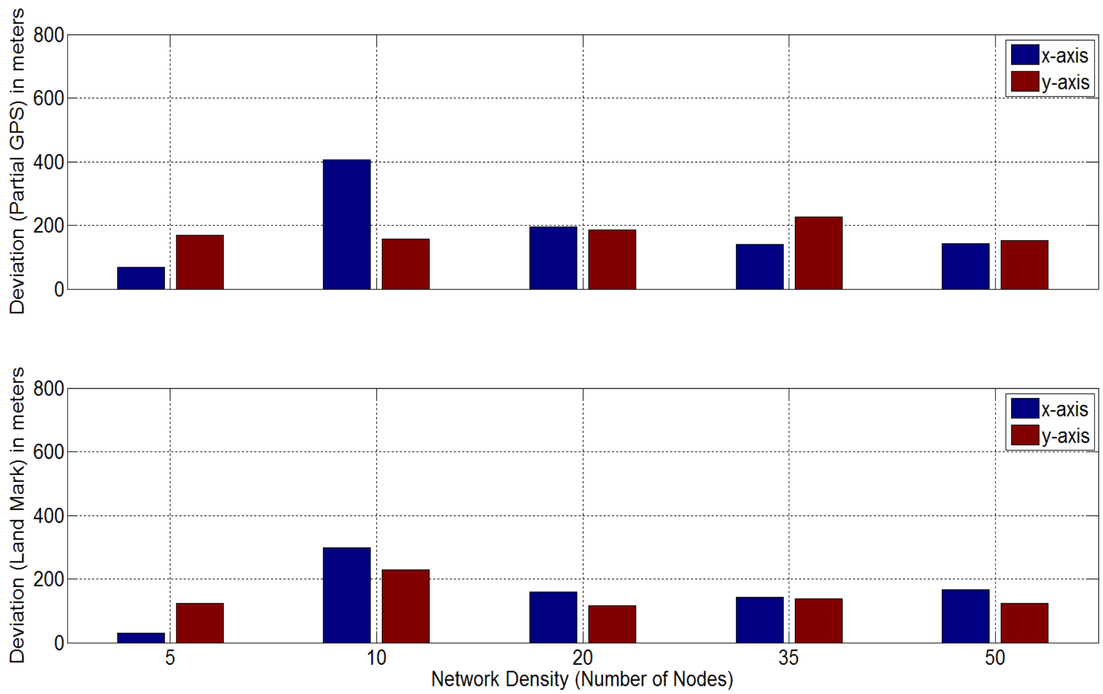


Figure 5-19: Approximate Localization for Static Scenario.

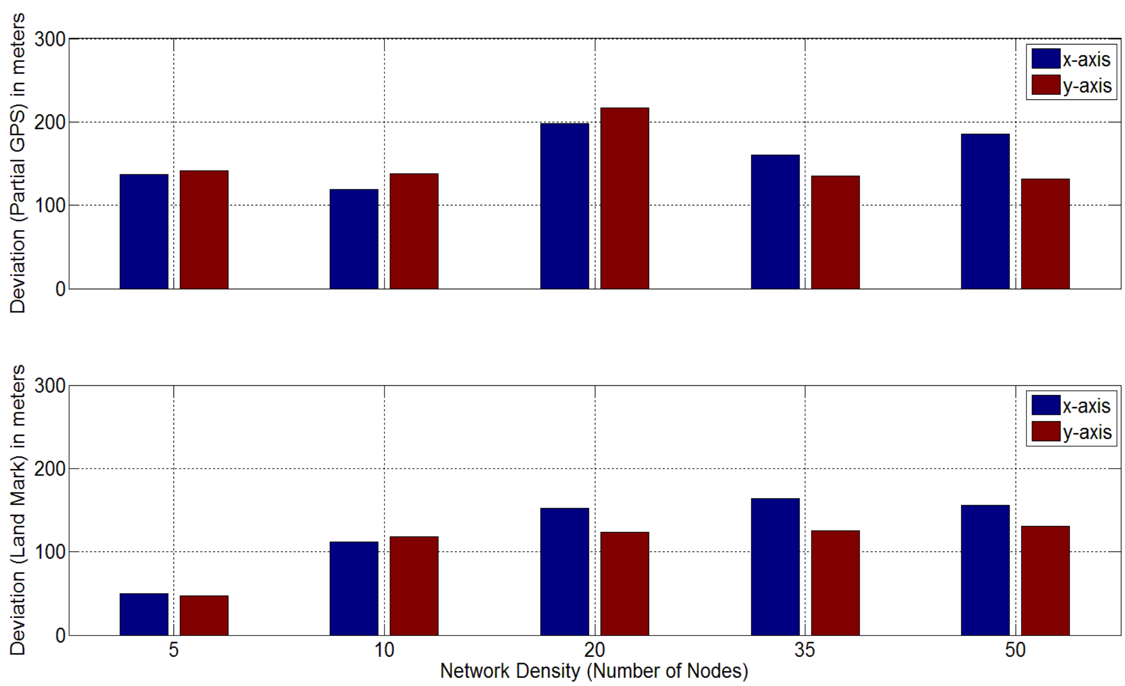


Figure 5-20: Approximate Localization with Random Walk 2D mobility model.

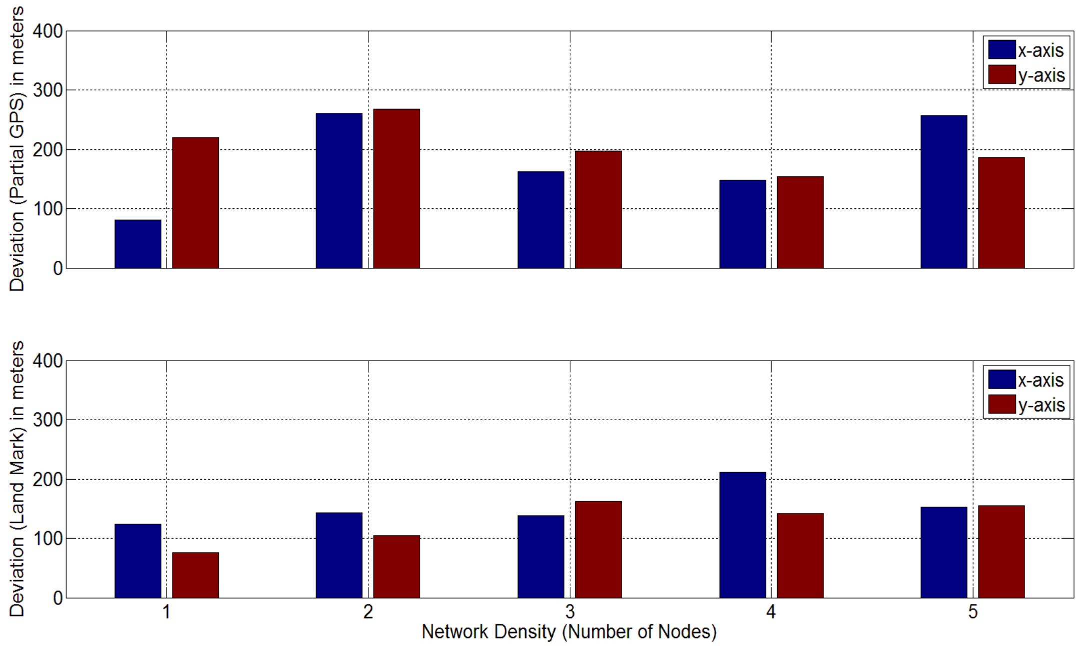


Figure 5-21: Approximate Localization with Random Direction 2D mobility model.

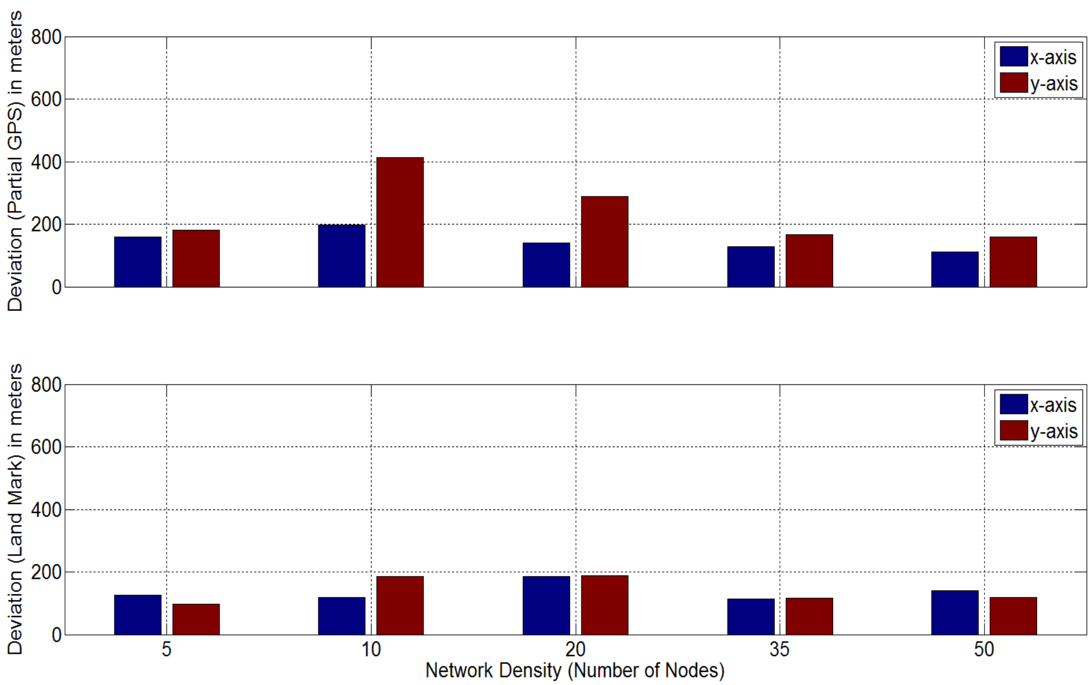


Figure 5-22: Approximate Localization with Random Waypoint mobility model.

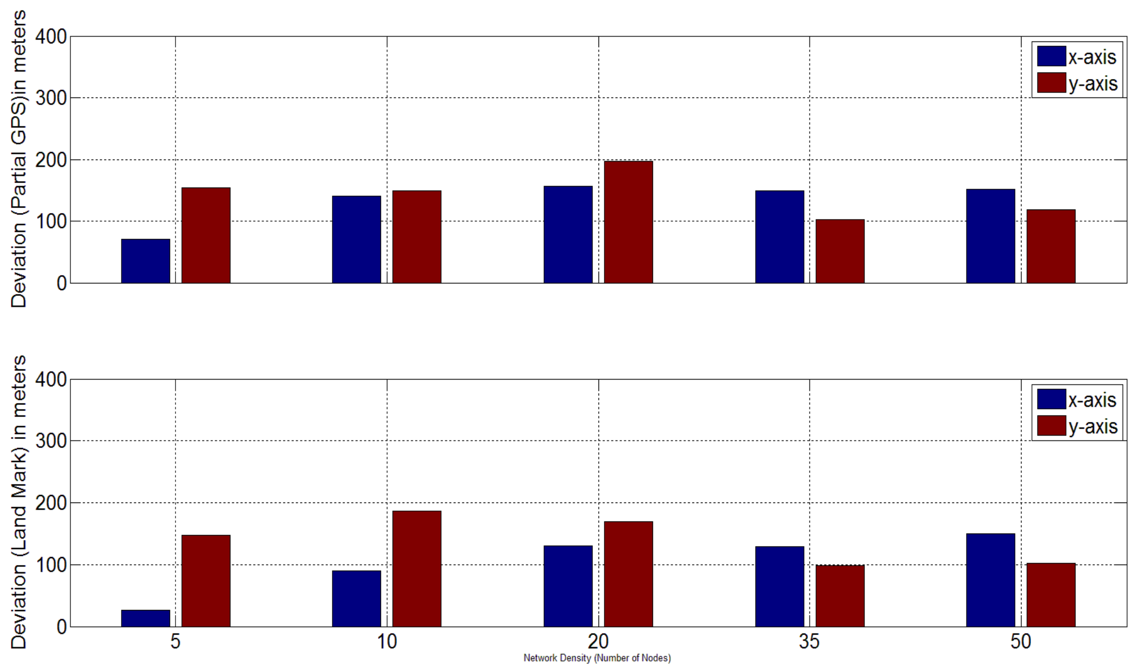


Figure 5-23: Approximate Localization with the Gauss Markov mobility model.

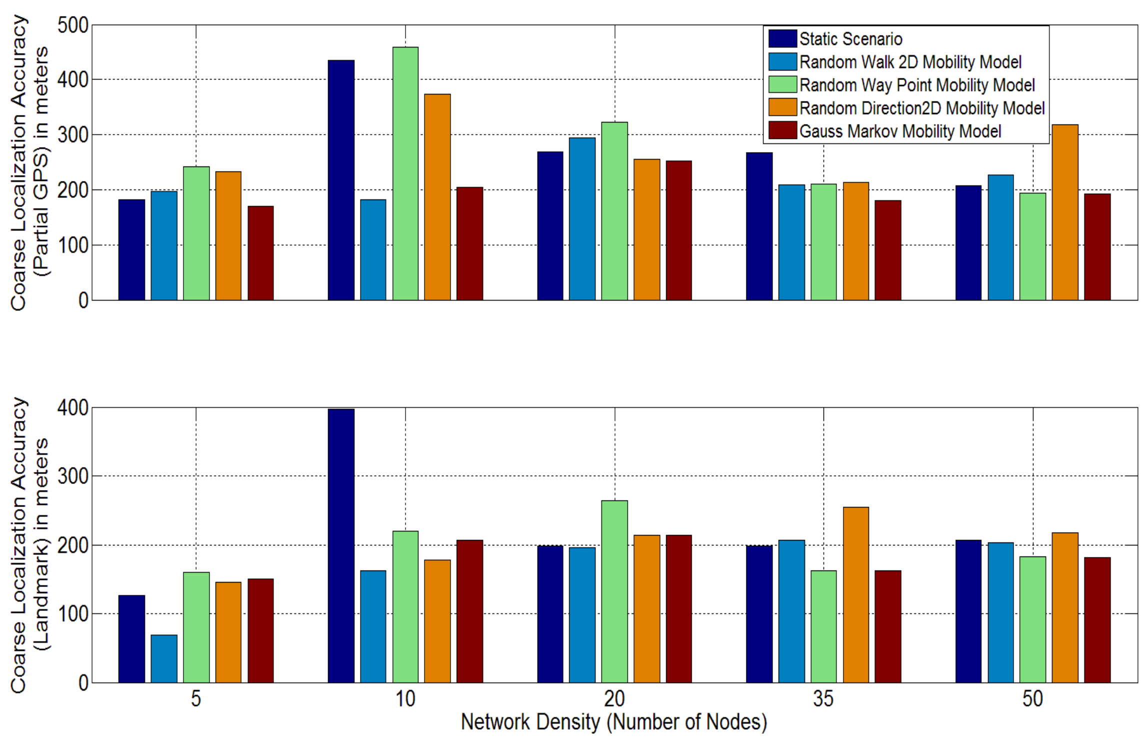


Figure 5-24: Coarse Localization Accuracy in terms of distance.

Figure 5-24 and Figure 5-25 shows the accuracy of our coarse localization approach in terms of distance from the original location. As shown in Figure 5-24, after providing only 25% nodes as GPS enabled nodes, the calculated locations are within 160 m to 460 m range but as we add 10% landmark information this distance drops within the range of 60 m to 390 m. As we can use any geographical information as landmark without any overhead compared to GPS, the landmark is increased from 10% to 25% on top of 25% GE nodes and the accuracy increase significantly and the distance range from 40 m to 190 m which is shown in Figure 5-25. In case of static scenario the localization error is higher than the mobile scenarios because the mobility actually helps layer-3 localization. In case of mobile scenarios layer-3 connectivity information are constantly changing as the nodes move and the number of undiscovered link will reduce as the disconnected moving nodes come within any other node’s transmission range. But in static scenario the number of undiscovered link remain same, as the disconnected nodes cannot come closer to the communication range of other nodes due to lack of movement.

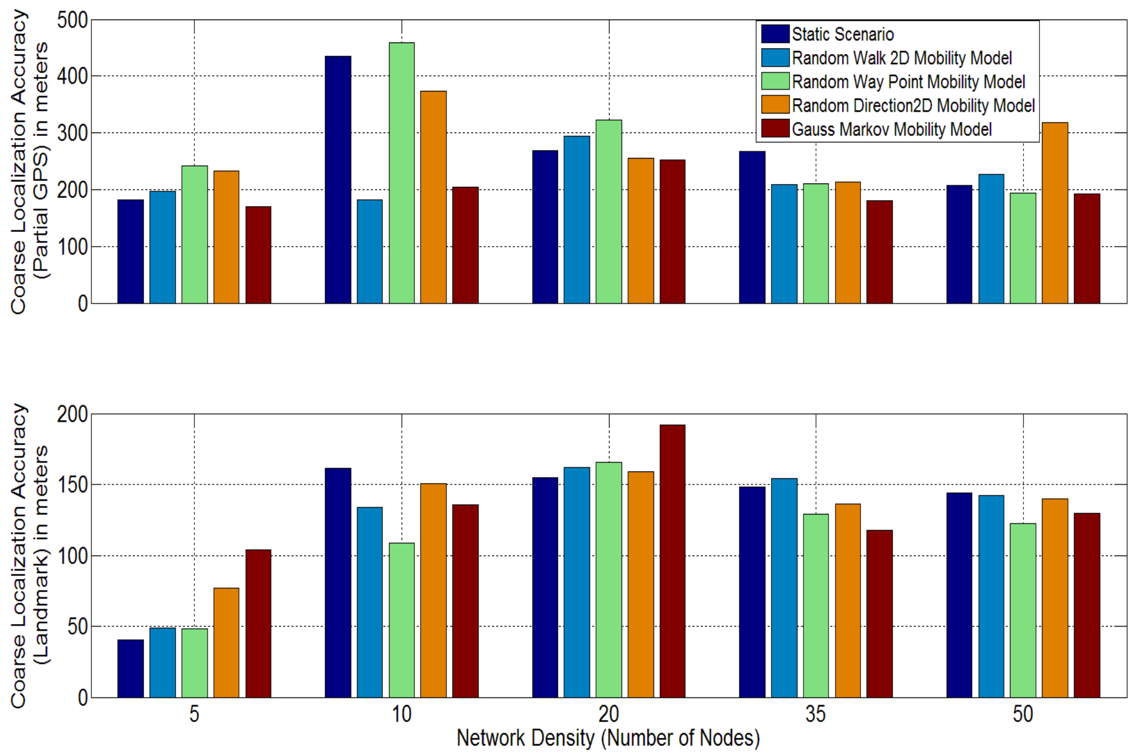


Figure 5-25: Coarse Localization Accuracy in terms of distance.

Chapter Six: CONCLUSION and FUTURE WORK

This thesis proposes a high level design for connectivity visualization of OLSR based TMANET topology based on local database available in an OLSR node for situational awareness system. The proposed approach takes advantage of the existing topology information exchanged among wireless nodes within the network. The proposed approach relies on the standard OLSR and the modification to the standard is not required.

We examined different approaches in using OLSR data for topology visualization, and showed the efficiency and accuracy of using OLSR databases and messages to build a topology map of the network. We further used simulations to study the performance of centralized versus nodal visualization and identified factors that contribute to errors in topology discovery.

We also come up with an analytical model to analyse the network bandwidth and overhead of using TC, HELLO and custom NIM message to evaluate the performance of centralized visualization to build full map of the network with respect to situational awareness system. Based on our findings there will additional 67 % to 75 % additional overhead if we use NIM message.

Finally we propose an algorithm based on force directed method to calculate the coarse location of the node. The key idea of this proposed approach that distinguish it from other localization algorithms based on force directed method is that it only relies on the OLSR topology information and as well as a small subset of GPs enabled nodes. To boost up the accuracy we also introduced landmark concept that decreases the calculated deviation from the original location dramatically.

However, this work is open to future development that may increase the topology discovery accuracy for better connectivity visualization and as well as may increase the accuracy of the localization calculation.

In current implementation of connectivity visualization of OLSR based TMANET topology we did not change any OLSR parameter as mentioned in the standard. As a future work, OLSR parameters can be optimised for better accuracy in topology discovery.

As we proposed a coarse localization approach we did not take into account received signal strength information or any other range based technique. These techniques can be added to our proposed algorithm in the future.

This thesis work can be continued by analyzing the impact of mobility parameters and patterns with respect to situational awareness systems, improving the topology discovery algorithm by approximating the node locations based on their connectivity and signal strength, and comparing OLSR with other MANET routing protocols in a situational awareness system.

APPENDIX A: NET VIZ

The network visualizer (code name: NetViz) was designed in an object-orientated fashion along with multiple threads.

- *PackedLinkLists.Java*

This is the main data structure that contains all links and nodes references. There are 4 internal data structures that watch and control all links and nodes. To improve search times for links and nodes we have introduced the dictionary data structure. For starters the dictionary map 'dictionaryLinks' keeps track of all links created between nodes. For every link there exist 2 keys; Node 1 + Node 2 and Node 2 + Node 1 where Node 1 and Node 2 are Strings of their ID's. If the key exists then the link exists. The dictionary map '_adjacentNodes' keeps track of all adjacent nodes connected to that key. This structure is used throughout theTopology thread.

- *CanvasThread.Java*

Netviz v1.4.0 at the time of documentation relies on 5 threads: Repaint; Topology; Position; Distance; and Reposition. Each one of these threads is orchestrated within CanvasThread.java. Each thread is thoroughly documented below:

  - *Thread Name: Repaint*

This thread simply repaints the canvas of the network visualizer. This thread updates every x number of milliseconds located in the settings.netviz file. This is a general refresh of the screen. On the other hand most fields contained within the canvas automatically recalls repaint when required. This is simply a failsafe for canvas repainting.

  - *Thread Name: Topology*

This thread updates the topology and location information of all nodes. This thread simply just updates the topology and location information based on 2 files; topology.netviz; nodes.netviz. An update occur every x number of milliseconds located in the settings.netviz file. When an update occurs the application locks the files using FileChannel to prevent any type of corruption throughout reading the files. The program simply searches line by line in each file searching for any changes. For instance throughout reading topology.netviz if there does not exist a previous link then we must introduce a new link. This thread before checking if the link exists, checks if the node exists and creates one when necessary. Throughout reading the file nodes.netviz the application changes that targeted node(s) with the information provided. When a node has information this node is anchored to its position.

- *Thread Name: Position*

This thread just updates the positions of all nodes when the system detects an unstable topology. This is where the main algorithm for repositioning nodes along with the general physics engine.

This is the physics engine thread which calculates all velocities for each node. This thread is also responsible for repositioning these nodes by a maximum velocity. This thread occur every x number of milliseconds located in the settings.netviz file. It first starts off by taking the summation of all axial dimensions for each node. In other words the dx and dy velocities are predetermined before repositioning all nodes. Once calculations are performed the thread continues onto repositioning nodes by looping through and taking a velocity into a direction vector.

- *Thread Name: Distance*

This thread creates and updates a file named distance.netviz which contains information about the distances of each node relative to one another.

This thread can be used to determine whether or not the graph produce is accurate to the true graphical layout of the adjacent matrix provided by topology.netviz. An update occur every x number of milliseconds located in settings.netviz file. When an update occurs the application locks the file using FileChannel to prevent a file corruption throughout modification of the file. This thread simply loops through all possible nodes and determines the theoretical distance between each node and creates a distance matrix. If the targeted node is connected to another node with a link then the value will be positive whereas on the other hand the value would be negative.

- *Thread Name: Reposition*

This is a temporary solution as this thread is not needed. The physics engine gets unwanted velocities due to accurate measurements and interfering velocities. As a result the graph traverses inappropriately away from the center of the canvas.

Essentially this thread is exactly as described: find the center of the graph and reposition all nodes closer to that position.

References

[1] Internet Engineering Task Force. Mobile Ad-Hoc Networks (MANET) - WORKGROUP. [Online]. http://www.ietf.org/html.charters/manet-charter.html

[2] J. Sun, "Mobile Ad-Hoc Networking: An Essential Technology for Pervasive Computing.," in *International conference on Info-Net*, 2001, pp. 316-321.

[3] J. Macker and M. S. Corson, "Mobile Ad-Hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations," draft-ietf-manet-issues-o1.txt, April, 1998.

[4] Dale Arden, Li Li, Paul Labbe, and Ying Ge, "self-Aware - Situational Aware : Integrated Handhelds for Dispersed Civil and Military Urban Operations," *Inside GNSS*, vol. 2, no. 2, pp. 34-45, March/April 2007.

[5] M. Gregoire and L. Beaudoin, "Visualization for Network Situational Awarness in Computer Network Defence," in *Visualization aand the Common operational Picture. Meeting Proceeding RTO-MP-IST-043*, Neuilly-sur-Seine, France, 2005, pp. 20-1-20-6.

[6] M. Gerla, "From Battlefields to Urban Grids: New Research Challenges in Ad-Hoc Wireless networks ," *Pervassive and Mobile Computing*, vol. 1, pp. 77-93, 2005.

[7] srinath Srinivasa et al., "A Symmetric Localization Algorithm for MANTETs Based on Collapsing Coordinate System," in *High Performance Computing HiPC*,

Heidelberg, 2005.

[8] Anand Dersingh, Z M Faizul Islam, Shahram Shah Heydari, and Md. Mustafizur Rahman, "OLSR-based Topology Discovery for Mobile Situational Awarness System," in *The 2nd International Conference on Ambient System, Networks and Technologies*, Niagra Falls, Canada, 2011.

[9] Raghavendra V. Kulkarni, Ganesh K. Venayagamoo, Ann Miller, and Chian H. dagli, "Network-centric Localization in MANETs based on Practical Swarm Optimization," in *IEEE Swarm Intelligence Symposiums.*, 2008.

[10] Zhonghai Wang and Sayed A. zekavat, "MANET Localization Via Multi-Node TOA-DOA Optimal Fusion," in *Military Communication Conference*, 2006.

[11] Amitangshu Pal, "Localization Algorithms in Wireless Sensor Networks: Current Approaches and Future Challenges," *Network Protocols and Algorithms*, vol. 2, no. 1, 2010.

[12] Akio Koyama, kenyu Kamakura, and Leonard Barolli, "MANET-viewer: A Visualization system for Mobile Ad-Hoc Networks," in *Proceeding of the 7th International Conference on Advances in Mobile Computing and Multimedia (MoMM 2009 Workshop)*, 2009, pp. 452-457.

[13] Deborah Estrin et al., "Network Visualization with Nam, the VINT Network Animator," *IEEE Computer*, vol. 33, no. 11, pp. 63-68, November 2000.

[14] Kurkowski, T. Camp, N. Mushell, and M. Colagrosso, "A Visualization and Analysis Tool for NS-2 Wireless simulations:iNSpect," in *Proceeding of the IEEE*

International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication System (MASCOTS), 2005, pp. 503-506.

[15] J. Mark Belue et al., "Research and analysis of Simulation-based Networking Through Multi-objective Visualization," in *Proceeding of the 40th Conference on Winter Simulation*, 2008, pp. 1216-1224.

[16] Georgia Institute of Technology. (2010, July) Georgia Tech Network Simulator (GTNets). [Online]. http://www.eee.gatech.edu/research/labs/MANIACS/GTNetS/

[17] A. karygiannis and E. Antonakakis, "mLab: A Mobile Ad-Hoc Network Test Bed," in *1st Workshop on Security, Privacy, and Trust in Pervasive and Ubiquitous Computing in Conjunction with the IEEE International Conference in Pervasive services*, July 2005.

[18] A. Karygiannis, E. Antonakakis, and A. Apostolopoulos, "Host-Based Network Monitoring Tools for MANETs," in *Proceedings of the 3rd ACM International Workshop on Performance Evaluation of Wireless Ad-Hoc, Sensor and Ubiquitous Network*, 2006, pp. 153-157.

[19] T. Clausen, P. Jacquet, and L. Viennot, "Investigating the Impact of partial Topology in Proactive MANET Routing protocols," in *5th International Symposium on Wireless personal Multimedia Communications (WPMC)*, 2002.

[20] T. Clausen and P. Jacquet. Optimized Link State Routing protocol (OLSR), RFC 3626. [Online]. http://hipercom.inria.fr/olsr/rfc3626.txt

[21] D. Niculescu and B. Nath, "Ad-Hoc Positionning System," in *Proceedings of*

INFOCOM, San Francisco, CA, 2003.

[22] Nagpal, H. Shrobe, and J. Bachrach, "Organizing a Global Coordinate System from Local Information on an Ad-Hoc Sensor Network," in *Proceedings of the Second International Workshop on Information Processing in Sensor Networks*, Palo Alto, April 2003.

[23] N. Bulusu, J. Heidermann, and D. Estrin, "GPS-less Low-cost Outdoor Localization for Very Small Devices," *IEEE Personal Communications*, vol. 7, no. 5, October 2000.

[24] N.B. Priyantha, H. Balakrisnan, E. Demaine, and S. Teller, "Anchor-free Distributed Localization in Sensor Networks ," in *1st International Conference on Em-beeded Networked Sensor Systems*, 2003, pp. 340-341.

[25] S. Capkun, M. Hamdi, and J.P. Hubaux, "GPS-free Positining in Mobile Ad-Hoc Networks," in *Proceedings of Hawaii International Conference on System Sciences*, 2001.

[26] Sungil Kim, Yoo Chul Chung, Yangwoo Ko, and Dongman Lee, "A Reactive Local Positioning System for Ad-Hoc Networks," in *IEEE Internationql Conference on Sensor Networks, Ubiquitous and Trustworthy Computing*, 2008.

[27] Alon Efrat, David Forrester, Anand Iyer, and G. Stephen Kobourov, "Force Directed Approaches to Sensor Localization," in *8th Workshop on Algorithm Engineering and Experiments*, 2006, pp. 108-118.

[28] Lei Luo and Adarshpal S. Sethi, "Topology Discovery in Wireless Ad-Hoc

Networks," Dept. of Computer and Information science, University of Delaware, Newark, DE 199716,.

[29] (2011, September) United States Army Research Laboratory. [Online]. http://www.arl.army.mil

[30] P. Jacquet, A. Laouiti, P. Minet, and L. Viennot, "Performance Analysis of OLSR Multipoint Relay Flooding in Two Ad-hoc Wireless Network Models," INRIA Rocquencount, France, INRIA Research Report N:4260, September, 2001.

[31] T. Clausen, P. Jacquet, and L. Viennot, "Comparative Study of Routing Protocols for Mobile Adhoc Networks," in *Med-Hoc-Net'02*, Sardegna, Italy, September, 2002, p. 10.

[32] J. Moy. OSPF version 2, RFC 2328. [Online]. http://tools.ietf.org/html/rfc2328

[33] A. Huhtonen, "Comparing AODV and OLSR Routing Protocols," in *Seminar on Internetworking*, Sjokulla, April, 2004.

[34] University of Southern California Information Science Institute , RFC 791 - Internet Protocol, Standards Track edition, September, 1981.

[35] S. Deering and R. Hinden, RFC 2460 - Internet Protocol, Version 6 (IPv6) Specification, Standards Track Edition, December, 1998.

[36] Thierry Plesse et al., "OLSR Performance Measurement in a Military Mobile Ad-Hoc Network ," in *IEEE Proceedings of the 24th International Conference on Distributed ComputerSystem Workshops*, 2004.

[37] P. E. Villanueva-Pena and T. Kunz, "Extending Network Knowledge: Making

OLSR a Quality of Service Conductive Protocol," Carleton University, Technical Report January, 2006.

[38] P. E. Villanueva-Pena, T. kunz, and P. Dhakal, "Extended Topology Knowledge for Localization and Quality of Service with OLSR," in *IEEE international Conference on Wireless and Mobile Computing, Networking and Communications (WiMOb)*, 2006.

[39] H. Tang and M. Salmanian, "Lightweight integrated Authentication for Tactical MANETs," in *The 9th IEEE International Conference for Young Scientists*, 2008, pp. 2261-2271.

[40] B. Q. Nguyen, "A Tool for Visualizing Mobile Ad-Hoc Network Topology Definition Files," US Army research Laboratory, Adelphi, MD, Technical report ARL-TR-350, February, 2006.

[41] G. Di Battisa, P. Eades, R. Tamassia, and I. G. Tollis, "Graph Drawing: Algorithoms for the Visualization of Graphs," Prentice Hall, Upper Saddle River, NJ, 1999.

[42] J. Chandran, "A Robust Frame of WSN Utilizing Localization Technique," *Computer Engineering and Intelligent Systems*, vol. 2, no. 7, pp. 36-45, April 2011.

[43] R. Davidson and D. Harel, "Drawing Graphs Nicely Using Simulated Annealing," in *ACM Transaction on Graphics*, 1996, pp. 301-331.

[44] D. Harel and M. Sardas, "Randomized Graph Drawing with Heavy Duty Preprocessing," *Journal of Visual Language and Computing*, no. 6 (3), pp. 233-253, 1995.

[45] W. T. Tutte, "Convex Representation of Graphs ," *Proceedings London Mathmetical Society* , vol. 10 (3), pp. 304-320, 1960.

[46] P. Eades, "A Heuristic for Graph Drawing," *Congressus Numerantium*, vol. 42, pp. 149-160, 1984.

[47] T. M. J. Fruchterman and E. M. Reingold, "Graph Drawing by Force-Directed Placement ," *Software Practice and Experience*, vol. 21, no. 11, pp. 1129-1164, 1991.

[48] T. Kamada and S. Kawai, "An algoritm for Drawing General Undirected Graphs," *Information Processing Letters*, vol. 31, pp. 7-15, 1989.

[49] P. Jacquet and L. Viennot, "Overhead in Mobile Ad-hoc Network Protocols," INRIA Rocquencourt, France, INRIA research report N:3965, June 2000.

[50] A. Agarwal and W. Wang, "Statistical Analysis of the Impact of Routing in MANET Based on Real-Time Measurements," in *ICCCN'05*, San Diego, California, USA, October, 2005.

[51] L. Viennot, "Complexity Results on Election of Multipoint Relays in wireless Networks," INRIA Rocquencout, France, INRIA research report N: 3584, December, 1998.

[52] T. Clausen, "Comparative Study of Routing Protocols for Mobile Adhoc Networks," INRIA Rocquencourt, France, INRIA research report N: 5135, March, 2004.

[53] Dang Nguyen and Pascale Minet, "Scalability of the OLSR Protocol with the Fish Eye Extension," INRIA Rocquencourt, France, HIPERCOM project report.

[54] (2011, December) Network simulator - NS-3. [Online]. http://www.nsman.org

[55] Mathieu Lacage and Thomas R. henderson, "Yet Another Network Simulator," in *Proceedings of the 2006 Workshop on ns-2: the IP Network Simulator* , Pisa, Italy, December October 2006. [Online]. http://cutebugs.net/files/wns2-yans.pdf

[56] (2011, January) NS-3 Logging Module. [Online]. http://www.nsnam.org/docs/tutorial/html/tweaking.html